AD-A246 706

COMPARISON OF THE OPERATIONAL
CHARACTERISTICS OF THE
THEORY OF CONSTRAINTS AND
JUST-IN-TIME SCHEDULING
METHODOLOGIES

THESIS

Lynn A. Sines, Captain, USAF

AFIT/GSM/LSM/91S-24

92-04901

DEPARTMENT OF THE AIR FORCE

AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

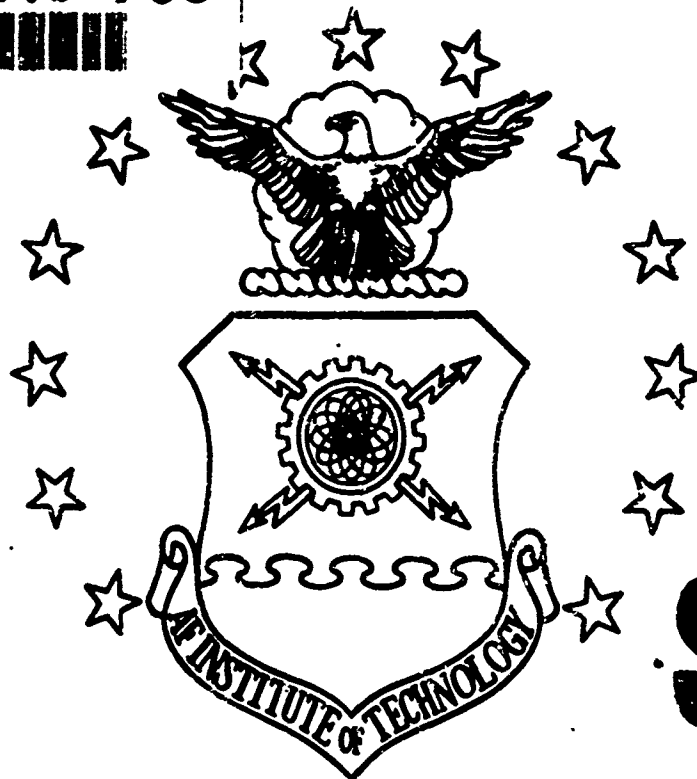Wright-Patterson Air Force Base, Ohio

92 2 25 086

AFIT/GSM/LSM/91S-24

DTIC
SELECTE
MAR 0 3 1992
S D
D

COMPARISON OF THE OPERATIONAL
CHARACTERISTICS OF THE
THEORY OF CONSTRAINTS AND
JUST-IN-TIME SCHEDULING
METHODOLOGIES

THESIS

Lynn A. Sines, Captain, USAF

AFIT/GSM/LSM/91S-24

The views expressed in this thesis are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

COMPARISON OF THE OPERATIONAL CHARACTERISTICS OF THE THEORY

OF CONSTRAINTS AND JUST-IN-TIME SCHEDULING METHODOLOGIES

THESIS

Presented to the Faculty of the School of Systems and

Logistics of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the

Requirements for the Degree of

Master of Science in System Management

Lynn A. Sines, B.S.

Captain, USAF

September 1991

Preface

The purpose of this thesis was to investigate the operational characteristics of Drum-Buffer-Rope, a Theory of Constraints scheduling methodology. The Theory of Constraints (TOC) has not been well researched in the past and this effort is one of the first evaluations of its operational characteristics.

The results of the simulation modeling were very encouraging, but some of the problems found in JIT systems are still present in TOC. This type of work with TCC should be continued so that when an organization implements the TOC control system they can be assured of the type of results expected.

In programming of the simulation models, running the models, and writing of this thesis I have had a great deal of help from others. I wish to thank my thesis advisor, Lt Col Richard Moore, for his patience and assistance when it was needed. I also wish to thank Dr. Fraker and the Computer Services Division of the University of Dayton for allowing me access to their computer and GPSSH software. Finally, I wish to thank my wife Kathy for her understanding on those nights when nothing was going right. I especially wish to acknowledge the help of my son Wil whose smiling face made each day a little brighter.

<div align="right">Lynn A. Sines</div>

<div align="center">ii</div>

Table of Contents

List of Figures

List of Tables

AFIT/GSM/LSM/91S-24

Abstract

This study compared the characteristics of scheduling using two approaches: Just-in-Time (JIT) and the Theory of Constraints (TOC). Computer simulation was used to evaluate changes in the throughput of the system due to system variability and varying work-in-process (WIP) levels. The independent variables were processing time, probability of failure, and WIP level. The literature search revealed that these variables impacted performance of JIT systems. Simulation models of three different production systems were developed to determine if the TOC scheduled system would provide greater or equivalent throughput for equivalent levels of WIP, and to determine if the system production flow path had an impact on the performance of the TOC system. The models were scheduled using both JIT and TOC methodologies. The mean throughput for each combination of independent variables was calculated along with the 95 percent confidence interval. The mean and confidence intervals were then graphed for analysis. Simulation results indicated that the TOC system outperforms the JIT system for single product systems while the results of the multiple product system were inconclusive. There was no difference between the TOC systems for process time variability, but there was a difference for process station failure.

xi

COMPARISON OF THE OPERATIONAL CHARACTERISTICS
OF THE THEORY OF CONSTRAINTS
AND
JUST-IN-TIME SCHEDULING METHODOLOGIES

I. Introduction

This thesis compares the performance of production systems which utilize Just-in-Time (JIT) and Theory of Constraints (TOC) scheduling methodologies. Dr. Eliyahu Goldratt, developer of the Theory of Constraints, has defined throughput, operating expense, and inventory as the three key performance measures in any system (Goldratt and Fox, 1986:28). The definitions of these three performance measures, as commonly used in American business, are:

1. Throughput – The total volume of production through a facility (Wallace and Dougherty, 1987:32).

2. Inventory – Items which are in a stocking location or work in process and which serve to decouple successive operations in the process of manufacturing a product and distributing it to the consumer (Wallace and Dougherty, 1987:15).

3. Operating Expense – Cost incurred in the course of the day-to-day activities of the firm (Skousen and others, 1987:1026)

The definitions Goldratt (1986:59-60) assigns to the measures above are simpler than those that are commonly used.

1. Throughput - The rate at which the system generates money through sales.

2. Inventory - All the money the system invests in purchasing things the system intends to sell.

3. Operating Expense - All the money the system spends in turning inventory into throughput. (Goldratt and Fox, 1986:29)

This research compares throughput and inventory of systems scheduled using TOC with the throughput and inventory of systems scheduled using JIT. Throughput will be the dependent variable of the model system. Inventory and system variability will be the independent variables for the comparison of JIT and TOC scheduling.

General Issue

Before the development of JIT, production scheduling was accomplished using heuristics to sequence jobs (Panwalker and Iskander, 1977:46). The first major step forward in scheduling was the Japanese Just-in-time (JIT) system. JIT provides an specific methodology for the scheduling of repetitive production systems which had not been provided by past methods. JIT allowed the Japanese to capture many markets due to improvements in quality, due date performance, and flexibility (Goldratt and Fox, 1986:36-60). While JIT was a major step forward in the development of scheduling methodologies, it has limited application outside of the repetitive manufacturing environment. In addition, it experiences a decrease in system reliability in the presence of variation in system

component reliability. The decrease in total system reliability can not be adequately protected against with increases in work-in-process (WIP) inventory as had been done in traditional scheduling methodologies (Lulu, 1986:237-238). The intent behind the JIT scheduling methodology was also to minimize the WIP inventory. Thus, while increased WIP inventory is required to protect the system reliability the increased inventory is in conflict with the intent of JIT (Schonberger, 1982:31-32).

Dr. Eliyahu Goldratt has said that the system variability problems associated with JIT can be alleviated through application of TOC to production control. Production organizations that implemented TOC have been able to maintain system reliability while decreasing WIP inventory below the levels of JIT (Goldratt, 1990:109-128).

United States industries must improve faster than the Japanese even to catch up, much less compete effectively in the world market. Through the initiatives currently under way in the U.S. Air Force, many organizations will compete with industry for workload and must be able to compete against companies employing JIT or TOC scheduled production. If TOC does perform to the level that has been claimed by its advocates, then Air Force organizations which utilize the TOC philosophy should be able to outperform commercial concerns which use JIT.

Specific Problem Statement

While JIT scheduling systems have been extensively investigated, TOC remains largely unresearched in the academic world. This difference in the amount of research that has been conducted is due primarily to the competitive threat of Japanese industry. Japanese industry has utilized JIT and other management methods to compete with American industry. The companies that have implemented the TOC philosophy have been primarily American or European. Academic research has been focused on JIT since it has been the scheduling methodology that has received the bulk of the business and public press attention. The large number of research efforts that were devoted to JIT may have impacted the number of available researchers that could have been evaluating TOC. Additionally, when compared to JIT, TOC concepts are relatively new. The first published explanation of the concept of drum-buffer-rope was not published until 1986. (Goldratt and Fox, 1986)

To develop realistic schedules and effectively control production in an actual organization, the effects of the scheduling theory on the organization must be known. To determine the suitability of the scheduling methodology research must be conducted to evaluate the expected impacts of the methodology.

Research Objectives

This research will investigate whether, through the use of TOC, system throughput is greater than that achieved with JIT scheduling techniques. A system that uses TOC scheduling should not be as significantly affected by system variabilities as one which uses JIT scheduling methodologies. The specific research questions that will be addressed are:

1. Does TOC's Drum-Buffer-Rope (DBR) technique resolve the throughput problems associated with system variability in JIT scheduled systems?

2. Is the throughput achieved in TOC systems dependent on the flow path of production system under consideration?

Scope of the Research

This research will be limited to the analysis of theoretical systems in which process component reliability and process time reliability are probabilistic variables. The total possible work-in-process inventory levels will be set at a constant level through the specification of buffer sizes and will only vary between simulation trials. The dependent variable in the system will be limited to the measurement of system throughput.

Background

Since the introduction of TOC in the early 1980s there has been little academic research conducted to evaluate the system effects of the TOC. Meanwhile a great deal of research has been conducted on JIT scheduled systems. These

1-5

articles have identified many problems with JIT which, if the claims of the proponents of TOC are true, make TOC a viable replacement for JIT. The most severe anomaly that affects production systems occurs when the system experiences variation in either process time or system reliability. Since JIT greatly reduces the work-in-process inventory that allows decoupling of operations, a small variation at a process station is magnified by the system and has a major impact on the ability of the system to produce an end item (Lulu, 1986:241).

II. Production Process Scheduling

Production process scheduling or control has long been recognized as a very important component in the effective utilization of a production system. Prior to the development of the Just-in-time (JIT) scheduling methodology efforts were made either to deal with the system on a gross capacity basis or in the sequencing of parts being fabricated. Both of these views of the system failed to incorporate the effects of interdependencies and inherent variabilities of the production system. This failure made the scheduling systems which resulted from the views very unreliable and operations managers viewed the generated schedules with suspicion. (Rowe, 1960:125)

In 1929, the Taylor Society acknowledged the problem of scheduling of a manufacturing concern. Control of the manufacturing process was viewed as essential to the smooth flow of processing and to meet sales demands. It was also seen that control of production and sales were correlated, but this correlation was viewed to add so much diversity that there was no way of establishing control over the productive process. So while establishing control was viewed as essential it was also viewed as impossible to accomplish. (Lansburgh, 1929:263-265)

The JIT philosophy was the first major effort which succeeded in establishing both required and realistic control over the production process. While JIT was the

first to succeed it was by no means the first effort made in establishing control. Prior efforts include:

1. Simple priority rules related to processing time, due date, number of operations, cost rules, setup time, arrival times, and machine selection.

2. Combinations of the simple priority rules above such as first in first out (FIFO), dividing jobs into priority classes and using FIFO, queuing rules, and cost/time criteria.

3. Weighted priority indexes which select the job with the smallest value of priority index.

4. Heuristic scheduling rules which include many different kinds of rules based on the operation under consideration.

All of these efforts failed to establish firm control over the production process due primarily to the subjective nature of the rules. JIT and TOC both establish firm control since they are based on the movement of subcomponents within the production system. (Panwalker and Iskander, 1977:45-59)

Just-in-time Scheduling Methodology

JIT mandates that the amount of work-in-process and finished inventory contained in the system should be very strictly controlled. JIT is both a system for managing an organization and an overall approach to production and inventory management (Heiko, 1989:61).

JIT, as an approach to production and inventory management, is used by many manufacturing companies in the United States and Japan (Walton, 1986:42). The basic idea of the production and inventory management portion of JIT is simply:

Produce and deliver finished goods just in time to be sold, subassemblies just in time to be assembled into finished goods, fabricated parts just in time to go into subassemblies, and purchased materials just in time to be transferred into fabricated parts. (Heiko, 1989:61)

Philosophy. The Japanese approach to the management philosophy of JIT is the elimination of waste or non-value-added actions. Taiichi Ohno states that JIT "is a system of production, based on the philosophy of total elimination of waste, that seeks the utmost in the way we make things" (Heiko, 1989:61).

The JIT management philosophy contains three primary guiding principles:

1. Streamline manufacturing operations. The streamlining of operations includes setting up dedicated lines and flow layout, and reducing or eliminating setup times (Heiko, 1989:62).

2. Control for total quality. Total quality control is the implementation of Dr. Edward Deming's 14 points of quality management (Heiko, 1989:62). These points are:

Point 1 - Create constancy of purpose toward improvement of product and service (Deming, 1982:23). The goal of the organization must be defined so that the entire organization can work toward the common goal (Walton, 1986:34).

Point 2 - Adopt the new philosophy (Deming, 1982: 23). The organization's management and workforce must adopt the attitude that mistakes that were tolerated in the past are now unacceptable (Walton, 1986:34).

Point 3 - Cease dependence on inspection to achieve quality (Deming, 1982:23). The organization tolerates the workers mistakes through the inspection of products. The organization, in a sense, pays for worker mistakes, then corrects the mistakes through rework. "Quality does not come from inspection but from improvement of the process" (Walton, 1986:34).

Point 4 - End the practice of awarding business on the basis of price-tag (Deming, 1982:23). The awarding of

2-3

business to a firm based upon price alone frequently leads to supplies being of low quality. The effective manager should seek the best quality, and work to achieve even better quality through one vendor under a long-term relationship (Walton, 1986:35).

Point 5 - Improve constantly and forever the system of production and service (Deming, 1982:23). Management should always look for ways to reduce waste and improve quality, This commitment should also be passed on to workers since they are closer to the problem areas (Walton, 1986:35).

Point 6 - Institute training on the job (Deming, 1982:23). Too often the workers can not do quality work because they have not been properly trained (Walton, 1986:35).

Point 7 - Institute leadership (Deming, 1982:23). Management should help workers do a better job. Management should also evaluate workers using objective methods to determine who is in need of individual help (Walton, 1986:35).

Point 8 - Drive out fear (Deming, 1982:23). Employees should not be afraid to ask questions or to take a position. To achieve improvements in quality and productivity, the people of the organization must feel secure (Walton, 1986:35).

Point 9 - Break down barriers between departments (Deming, 1982:24). All units in the organization must work together as a team and work toward the same goal (Walton, 1986:35).

Point 10 - Eliminate slogans, exhortations, and targets for the workforce asking for zero defects and new levels of productivity (Deming, 1982: 24). Slogans have never helped anyone do a better job (Walton, 1986:35).

Point 11 - Eliminate numerical quotas on the factory floor and eliminate management by objective (Deming, 1982:24). As Dr. Deming states, "Quotas take account of numbers, not quality or methods" (Walton, 1986:35).

Point 12 - Remove barriers to pride of workmanship (Deming, 1982:24). People want to do a good job, and get discouraged when the process does not allow quality. The normal barriers to quality are misguided supervisors, faulty equipment, and defective materials (Walton, 1986:36).

Point 13 - Institute a vigorous program of education and self improvement (Deming, 1982:24).

Management and workers must be trained in the new management methods such as teamwork and statistical techniques (Walton, 1986:37).

       Point 14 - Put everyone in the company to work to accomplish the transformation (Deming, 1982:24). Neither the workers nor management can implement the transformation on their own. All people in the organization must work together as a team (Walton, 1986:37).

       3. Leverage with worker creativity. Management must realize that the average worker is very creative and this creativity must be focused on the process. As in Deming's 14 points, the worker is closest to the process (Heiko, 1989:63).

While, the application of the JIT philosophy to the production system is extremely inexpensive, it does call for a change in the traditional culture of American business. The implementation of JIT does not require the use of computers or other capital investments, since it provides control over the inventory in the system through the use of visual indicators.

Kanban. Kanban is a Japanese word meaning "visible record" (Schonberger, 1982:86). The kanban can be anything from a golf ball to a card, but its basic function is to indicate to the receiver of the kanban that a quantity of parts is required elsewhere in the production process. The kanban functions in conjunction with the other elements of the JIT philosophy to provide production control and process improvement (Suzaki, 1987:148-149).

       1. Production control - The function is to tie different processes together and deliver the required materials at the right time in the right amount (Suzaki, 1987:152).

2. Process improvement - The emphasis is on reducing the inventory levels in the system through the removal of kanbans (Suzaki, 1987:152).

The production control aspect of the use of kanbans is utilized to provide a controlled environment to the production operation. The idea of the kanban is to link adjoining operations with a "rope". The rope is the combination of the card and parts containers. For example, if there are ten containers between two production processes each with the ability to hold one component then the work-in-process inventory level between the processes has been set at ten components. As the first operation completes processing of a component it is placed in an empty container. The first operation now checks to see if there are any remaining empty containers. If empty containers remain then work begins to process another component. If there are no empty containers then no processing takes place until a container is emptied by the second operation (Ricard, 1984:7). This movement of containers takes place between each set of production stations in the production process.

As can be seen by the discussion above the primary type of production operation to which the JIT with kanbans system could be applied is a flow-shop. A flow-shop and job-shop as defined by the American Production and Inventory Control Society are:

1. Flow-shop - A manufacturing organization in which machines and operators handle a standard, usually uninterrupted material flow. The plant layout is designed

to facilitate a product "flow." Production is set at a given rate and the products are generally manufactured in bulk (Wallace and Dougherty, 1987:12).

2. Job-shop - A manufacturing organization in which the productive resources are arranged according to function. The jobs pass through the functional departments in lots and each lot may have a different routing (Wallace and Dougherty, 1987:15,16).

The application of this type of control system to a job-shop is very complicated due to the number of different types of components and processes which are normally involved. The application of JIT with kanbans can be used in a job-shop in the area of raw material or subcomponent control and ordering (Schonberger and Knod, 1991:369).

The Theory of Constraints Scheduling Methodology

The overall objective of the Theory of Constraints (TOC) is to implement a process of continuing improvement. The continuing improvement process is achieved through commitment of an organization's management to the theories and practices that comprise TOC (Goldratt and Cox, 1986:266-272).

TOC is based on the theory of Dr. Eliyahu Goldratt which builds on the principles of the Quality Management Theory developed by Dr. Edward Deming, and the Just-in-time (JIT) inventory control procedure developed by Taiichi Ohno (Goldratt, 1990a:109).

The Theory of Constraints is a methodology for analyzing a system to determine the location and type of constraints present. Prior to the constraints being

2-7

analyzed, the management of the organization must determine
the organization's goal and the measurement criteria
applicable to those goals. The goals of the organization
will drive the decisions of how to manage the constraints.
After the goals and the measurement criteria are determined,
management can begin the process of analyzing the
constraints (Goldratt, 1990a:3-4). There are five
facilitating steps of TOC:

Step 1 - Identify the system's constraints. The
constraints must be identified and then prioritized
according to their effect on the system (Goldratt, 1990a:5).
Constraints can be categorized as either physical
constraints or policy constraints (Goldratt and Fox,
1989a:2).

1. Physical constraints. Physical constraints
are those physical things in the system which limit the
production capability. They can be resources that are in
short supply, vendor limitations, the demand for the
product, or long processing time operations which affect the
flow of material internal to the system (Goldratt and Fox,
1989a:2-4).

2. Policy constraints. Policy constraints
include all rules levied on the organization by management
that limit the system (Goldratt and Fox, 1989a:5-7).

Step 2 - Decide how to exploit the system's
constraints. The system's constraints should be fully
exploited so that the maximum output is achieved from the
system. Non-productive time for the constraint is the same
as non-productive time for the entire system (Goldratt,
1990a:5). On the surface it appears that the easiest type
of constraint to exploit is a policy constraint since the
policies that govern an organization are controlled or
influenced by management, but these constraints are the
least obvious since they are imbedded in the structure of
the organization (Goldratt and Fox, 1989a:3; Goldratt and
Fox, 1989b:22-23).

Step 3 - Subordinate everything else to the above
decision. The decision of how to exploit the full potential
of the constraint will place limits on all other decisions
affecting operation of the system. The constraint is the

limiting factor in the performance of the system (Goldratt, 1990a:5).

Step 4 - Elevate the system's constraints. If we can elevate or reduce the impact of the constraints' limiting effect, the system's performance can be improved. If the constraint can be sufficiently elevated the constraint will eventually be broken (Goldratt, 1990a:5-6).

Step 5 - If in the previous steps a constraint has been broken, go back to the first step. The constraint that has been broken in the preceding steps will no longer be the limiting factor in the system, but there will be other constraints on the system (Goldratt, 1990a:6).

The preceding five steps used in analyzing and resolving constraints are all equally important but, as Dr. Goldratt warns, the effective manager must "not allow inertia to cause a system constraint" (Goldratt, 1990a:6).

Scheduling with the TOC. Several of the tenets of TOC run counter to the conventional view of a production organization. These tenets are:

1. Balance flow, not capacity. Traditional manufacturing control systems have tried to balance capacity and then maintain flow. While the theoretical "perfect system" capacity can be balanced, the inherent variability of the real environment makes this impossible (Berry, 1985:81).

2. The level of utilization of a non-constraint is not determined by its own potential but by some other constraint in the system (Berry, 1985:81). As stated in the Theory of Constraints, the constraints of the system drive the performance of all aspects of the system (Goldratt, 1990a:4).

3. Utilization and activation of a resource are not synonymous. Management should not strive to keep everyone busy all the time. The only resource that should be 100 percent utilized is the constraint of the system (Berry, 1985:81).

4. An hour lost at a constraint is an hour lost for the total system. Since the constraint controls the

system it must be active constantly for the maximum system performance to be achieved (Berry, 1985:81).

5. An hour saved at a non-constraint is a mirage. A non-constraint is defined as having excess capacity, so time saved on it is only an increase in excess capacity (Berry, 1985:81).

6. Constraints govern both throughput and inventories. The amount that the system can produce (throughput) and the amount of material contained in the system (inventory) are controlled by the constraint. Thus, the constraint is the controlling element in the system (Berry, 1985:81).

7. The transfer batch may not, and many times should not, be equal to the process batch. The transfer batch must be sized to keep the constraint fully utilized and not by the theoretical economic process quantity (Berry, 1985:81).

8. The process batch should be variable, not fixed (Berry, 1985:81). The principles of JIT should be applied to minimize the amount of inventory contained in the system (Heiko, 1989:62).

9. Schedules should be established by looking at all the constraints simultaneously. The capacity of the constraints specifies the timing of the schedules and the ability of the system to produce (Berry, 1985:81).

Under TOC there are essentially three ways to develop a realistic production schedule, Optimized Production Technology (OPT), DISASTER, and Drum-Buffer-Rope (DBR), all of which use the TOC philosophy. The first developed by Dr. Goldratt was the Optimized Production Technology (OPT) scheduling software. In the OPT software the user constructed the flow of the organization, then input the processing times, market requirements, and all other production data. The software then generated a complete schedule for the organization which could be updated in a matter of minutes as compared to the hours required to

regenerate an MRP schedule. One of the problems associated with the OPT software was that it required a mainframe computer which was available for the large corporations which initially implemented the software, but was not available to numerous small businesses. The DISASTER software provided a Personal Computer based scheduling system to the small company. DISASTER is a computer-based implementation of Drum-Buffer-Rope (DBR). DBR is a manual, philosophical-based approach to the problem of scheduling that is simple enough to implement in many cases without the need for computers. (Lundrigan, 1986:5-9; Severs, 1991:4-5)

Drum-Buffer-Rope (DBR). Just as in JIT where kanban provides the primary means of control over the production system, DBR provides the same type of control in the TOC philosophy. While being similar to kanban, DBR is much easier to implement since the control is tying the constraint of the system to the material release point for the system (Goldratt and Fox, 1986:94). The idea of DBR comes from an analogy that Dr. Goldratt developed in The Goal (Goldratt and Cox, 1986) of a Boy Scout troop marching in single file on a hike.

In the hike analogy the drum is the slowest scout since he sets the pace for the troop. The rope is tied from the slowest scout to the first scout in the line. The buffer is the amount of slack that is placed in the rope to account for variations in the speed of the scouts. The scouts in the line after the slowest are constrained to travel no

faster than the slowest scout since the scout master has specified that there will be no passing. (Goldratt and Fox, 1986:96-99)

The hike analogy can also be applied to the JIT system. In a JIT system the rope is tied between each scout in the line and the buffer between each scout is the amount of slack in the particular segment of rope. The drummer of the system is the last scout in the line or in a production system the market demand for the product produced. (Goldratt and Fox, 1986:94-95)

In a DBR controlled production system the amount of buffer required is governed by the amount of variability or unreliability in the operations preceding the system constraint. The objective of the controlling process is the maintain 100 percent utilization of the constraint. The utilization of the other processes in the system is much less important since by definition they have excess capacity. (Goldratt and Fox, 1986:100-103)

The overall point to DBR control is that material is not released to supply work to workers, but is released and processed according to the needs of the system constraint. This idea of having idle labor is a major change to the culture of most American companies where in the past the main duty of management was to keep everyone busy all the time. (Goldratt and Fox, 1986:112-115)

Advantages of the TOC. The TOC has been successfully implemented by several of the major manufacturing companies

in the United States in the last 10 years. The initial implementation of the TOC was in producing prefabricated chicken coops in Israel. From this humble beginning, US companies such as General Motors, General Electric, AVCO, Ford, and Bendix have implemented the TOC concepts and achieved remarkable results. The results achieved include decreased inventories and lower processing times (Jayson, 1987:18).

The primary advantage of the TOC is the new way at viewing the organizational environment (Waters, 1986:107). In the past, manufacturing managers have tried to achieve local optimization by, as Dr. Goldratt states, thinking that "the only way to reach a global optimum is by insuring local optimums." The local optimization led to nothing more than clogging the constraints of the system and, as the managers tried to balance the production, the bottlenecks wandered from station to station. Thus the motto of the TOC is: "The sum of the local optimums is not equal to the global optimum" (Berry, 1985:81-82).

Another advantage of TOC and particularly the Drum-Buffer-Rope control methodology is the ease in implementing the system. While JIT is primarily constrained to the flow shop environment, DBR can be implemented in either a flow shop or job shop. Since the primary concern of the control system is to keep the constraint fully utilized and not controlling between each production station the mixture of

products going through the system can be much larger
(Goldratt, 1990b:222-228).

III. Methodology

Method of Approach

System simulation will be used in this investigation to
determine the effects of system variability on JIT and TOC
scheduled systems. In the JIT systems kanban control will
be used, while the TOC systems will use drum-buffer-rope
(DBR) control. The variability of the production systems
will consist of process time variation as well as breakdowns
resulting in downtime. Theoretical production systems with
different flow paths will be scheduled using both the JIT
and TOC scheduling methodologies to determine if the type of
production system effects the selection of optimum
scheduling methodology.

Background - Simulation of Production Systems

Past production system scheduling research has
extensively utilized simulation to determine the effects of
JIT scheduling on the system. The effects of system
variability on JIT scheduled systems, particularly
variability caused by process failure, has been shown to
degrade the ability of the system to produce at a constant
rate (Lulu, 1986:248; Lulu and Black, 1987:227-228). This
type of variability will be simulated in this study to
determine if these effects are also found in TOC scheduled
systems.

There have been no past studies that have compared JIT and TOC. The only comparative studies that were found were between JIT and traditional scheduling methodologies (Pyke and Cohen, 1990; Spearman and Zazanis, 1988; Austin and Khoshnevis, 1986; Karmarkar, 1986). In these studies the analysis of the optimum methodology was based primarily on the qualitative assessment of system performance by the researcher. The main problem with providing a quantitative comparison between traditional and JIT comes from the inherent differences in the operating philosophies behind the methodologies. Under traditional scheduling there is no effort to limit the amount of inventory in the system. In contrast, limiting inventory is a primary factor in the JIT philosophy. The quantitative studies that have been conducted have dealt primarily with the assessment of the operational characteristics of JIT. (Manivannan and Pegden, 1990; Ketcham, 1988; Lulu and Black, 1987; Lulu, 1986; Schroer and others, 1985; Huang and others, 1983) The studies that have been conducted on various configurations of JIT systems all conclude, that without the variability reductions, inherent in the JIT philosophy, the ability of the system to produce the intended output is severely impacted.

Justification of the Approach

Due to the complexities of production environments and the costs associated with varying actual systems, simulation

is the most feasible approach for the study of these systems. Also due to the variability of the human element in actual systems precise mathematical measurements are either infeasible or costly to obtain (Malcolm, 1958:177-178).

The use of simulation to study production systems has been acknowledged to be not only the best means for researching the effects of changes to the system, but also in most cases it is the only reasonable means to study such effects. Experimentation on the actual system is disruptive to the production environment and normally not cost effective (Law, 1988:40). The results obtained from experimentation on the actual system may also yield results that are affected by extraneous variables that can not be controlled. Through the use of simulation the total environment is controllable and the effects of extraneous environmental variables can be eliminated (Dunham and Kochhar, 1983:19). In the business world today simulation is routinely used for day-to-day decision-making and long-range planning. Some of the typical applications include inventory management, scheduling, and operations planning (Russell and others, 1967:162-163).

Assumptions

The following assumptions have been made relating to the simulation of these production systems:

1. The processing time for each production station is uniformly distributed.

2. Processing time and probability of process station failure will not both vary in the same simulation run.

3. Probability of process station failure is uniformly distributed with respect to time.

4. The probability of failure of the system constraint is zero.

5. Each buffer within the particular system is of the same capacity.

6. The market for the products produced can absorb the total throughput of the system.

General Research Method

This research study will utilize three different theoretical production systems to determine if throughput variations due to variations in work-in-process inventory, probability of process station failure, and process time variation are system dependent. Each of the production systems will be scheduled using JIT (Kanban) and TOC (Drum-Buffer-Rope) scheduling techniques to determine if throughput level is affected by scheduling methodology when subjected to system variations.

Specific Methodology

Development of Theoretical Production Systems. The research study will utilize three theoretical production systems to compare the operational characteristics of the TOC with JIT under a variety of conditions. The three types of production systems that will be utilized are:

Model System 1 - Single Product Assembly Line. This system will consist of a single product manufacturing

line with no parallel actions required to complete the product (Lulu, 1986:237). The actions required to complete the product will be performed by four process stations in a serial network (Figure 3-1). This type of production system is similar to type of system that aircraft engines are repaired under at an Air Force Air Logistics Center. The four steps in the actual system could be: (1) Incoming Inspection, (2) Engine Disassembly, (3) Engine Repair, (4) Engine Reassembly.
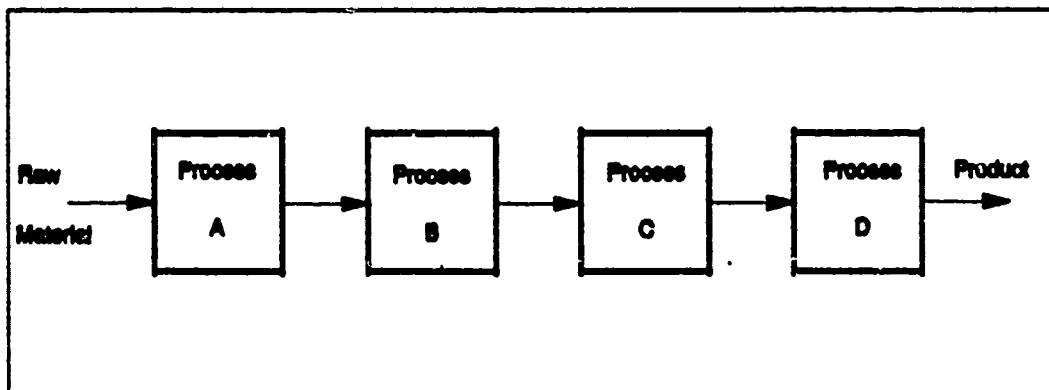
Figure 3-1. Single Product Assembly Line

Model System 2 - Single Product Parallel Process Assembly Line. This system model will consist of a total of six process stations. The raw material required by the product will flow through either of two parallel paths, each consisting of two process. The two subassemblies are then combined and production of the finished component is completed by the two remaining processes (Figure 3-2).

3-5
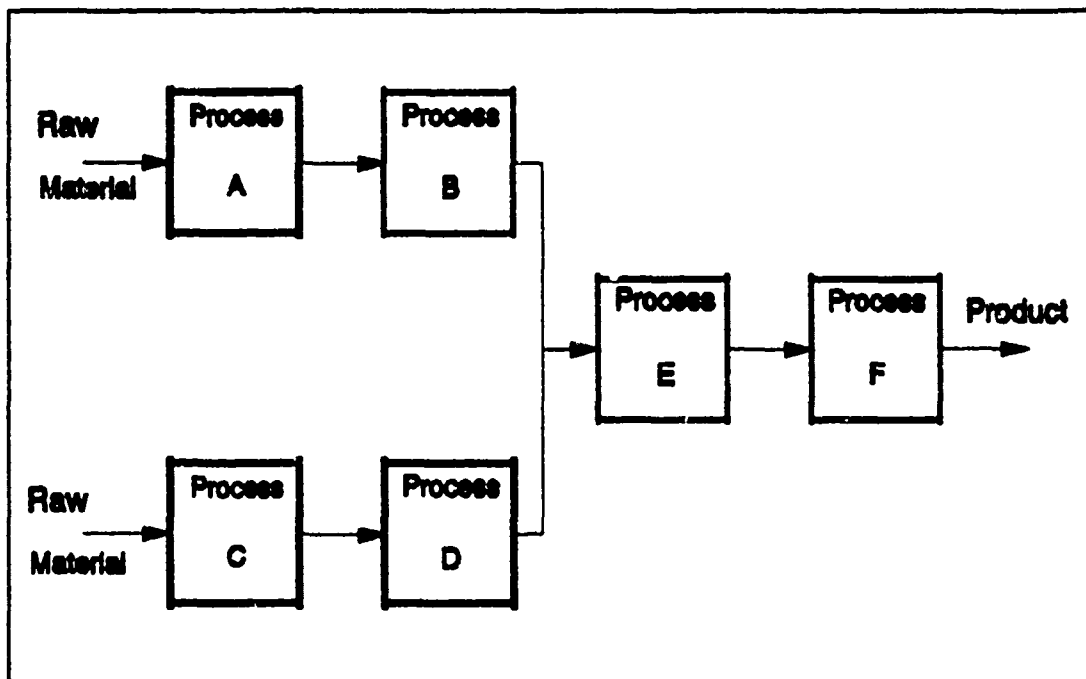
Figure 3-2.   Single Product Parallel Process Assembly Line

The single product parallel process system is analogous
to the type of systems found in defense contractor
facilities.   The two subcomponents that are combined into a
finished product could be the F110 engine and F-16 airframe
in the production of the F-16 fighter aircraft.

Model System 3 - Multiple Product Parallel Process
Assembly Line.   This system model will produce two different
products with parallel processes required to complete both
products.   Each product will consist of two subassemblies,
one of which is common to both products.   Each of the three
subassemblies will flow through two processes before being
combined to flow through the remaining two process cells to

be completed into a finished product. The completion of the
products will not require the utilization of the same
production assets and material to be completed (Figure 3-3).

The system could be the production system used in the
manufacture of aircraft pneumatic regulators. Two different
regulators are produced from one unique component, the
regulator body. and one common component, the actuating
mechanism. The regulator body and actuating mechanism are
then combined into the finished product. Pneumatic
regulators are found on all Air Force aircraft.
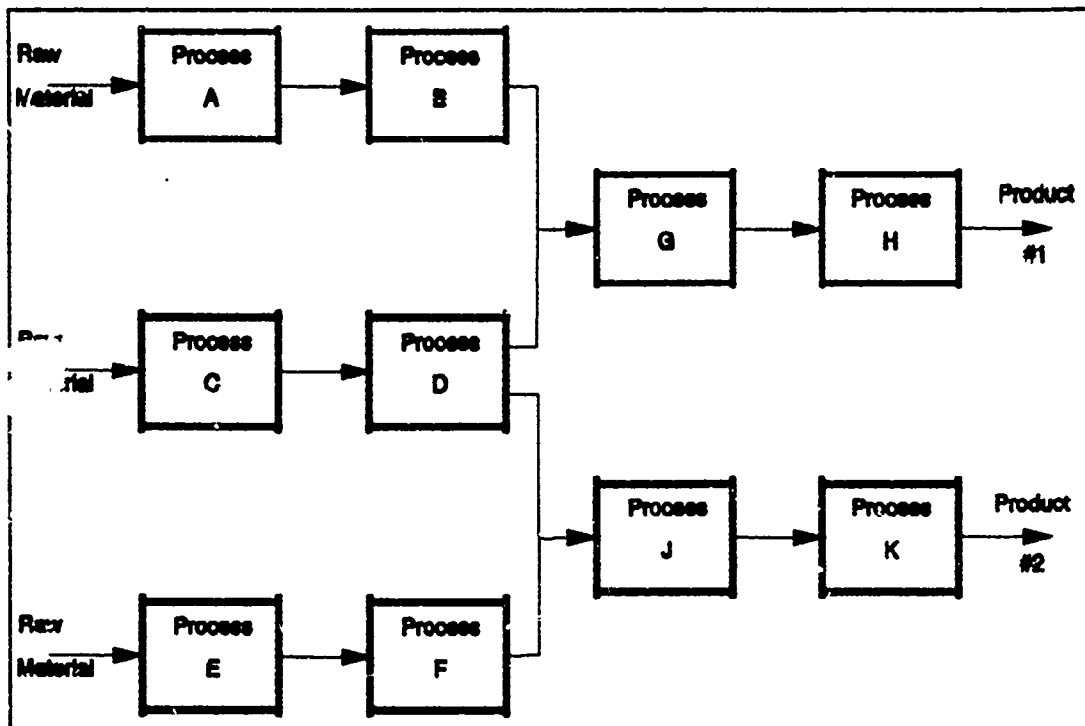
Figure 3-3.   Multiple Product Parallel Process
Assembly Line

System Model Development

Simulation Language. General Purpose Simulation System (GPSS/H) (Wolverine Software Corp., 1990) will be utilized as the simulation language to develop computer models of the above three systems. This simulation language is flexible enough to model a wide range of systems, but also provides the necessary capability for this application. GPSS/H has also been shown to be very reliable in a variety of applications (Banks and others, 1989:ix). In past research GPSS has been used effectively in the study of work-in-process inventory in a production system (Bowen, 1970). The past usage of GPSS combined with the inherent speed and flexibility of the simulation language make it appropriate for this research application.

System Model Programming. The system models were developed and programmed using the simulation language. Models were developed for each system using both JIT (JIT with kanban) and TOC (Drum-Buffer-Rope) scheduling methodologies. The symbols used to describe the buffer, production station, product flow, and control signal flow in the production systems are shown in Figure 3-4. These symbols will be used to describe the general flow through the systems that the programming will follow. The models will be developed from these general flow diagrams.

```
┌─────────────────────────────────────────────────┐
│                                                  │
│   ▦▦▦        BUFFER              ──▶   PRODUCT    │
│   ▦▦▦                                  FLOW      │
│                                                  │
│                                                  │
│   ┌──┐      PRODUCTION                 CONTROL   │
│   │  │      STATION             ◀──    SIGNAL    │
│   └──┘                                  FLOW     │
│                                                  │
└─────────────────────────────────────────────────┘
```
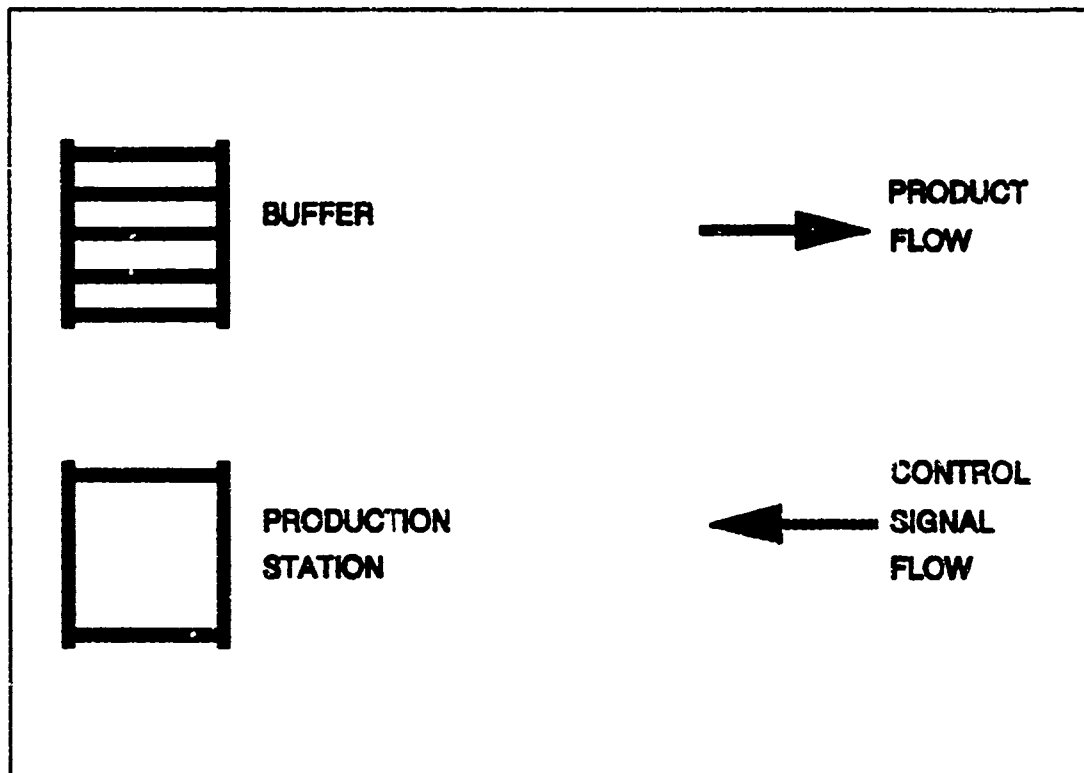
Figure 3-4.   Production System Symbols

The programming of System 1 - Single Product Assembly Line for the JIT with kanban model was developed with buffers located before each production station. The kanban control signal and product flow will be as illustrated in Figure 3-5. The work-in-process inventory was measured from the time the product entered Station 1 until the product completed processing on Station 4. This flow diagram illustrates a general production line controlled by the JIT methodology. The simulation program will duplicate this type of system flow pattern.
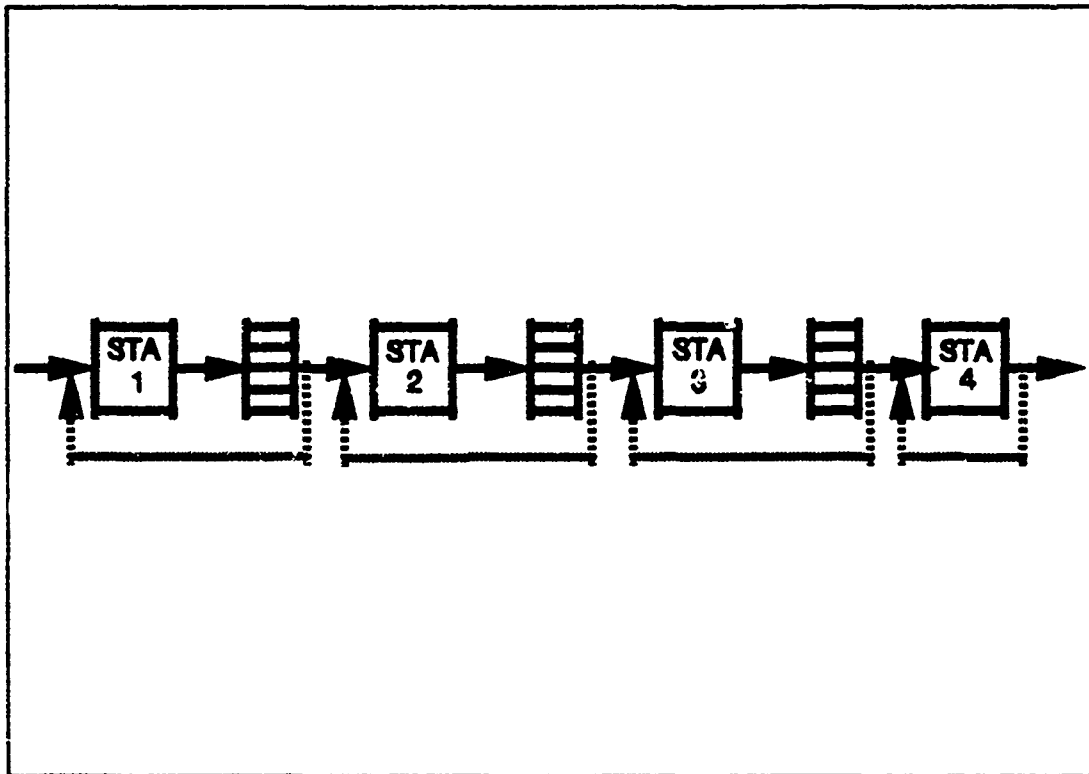
Figure 3-5.   System 1 - JIT with Kanban

The programming of System 1 using TOC Drum-Buffer-Rope control was developed with the buffer protection immediately before the system constraint (Station 3). The control signal flowed back to the initial product release point for r .ease into Station 1 (Figure 3-6). The work-in-process inventory was measured, as in the prior model, from the time the product entered Station 1 until it departed Station 4. The programming of the actual model followed the general flow pattern established by this diagram. The control of the model was established through the use of a buffer counter.
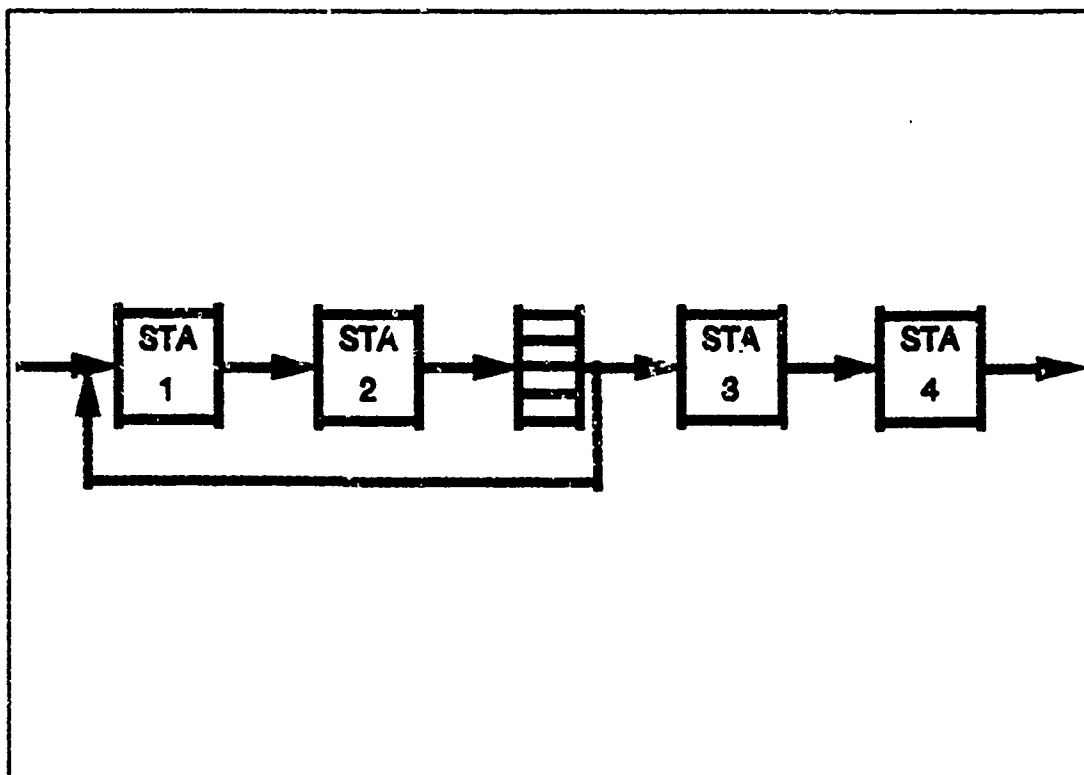
Figure 3-6. System 1 - Drum-Buffer-Rope

The programming of System 2 using JIT with kanban followed the same buffering and kanban flow as System 1 using JIT with kanban control. The work-in-process was measured by taking the amount of product in each of the parallel operations as half of a unit. The buffers, however, considered a subcomponent as a whole unit. The flow for the programming of the model is illustrated in Figure 3-7.
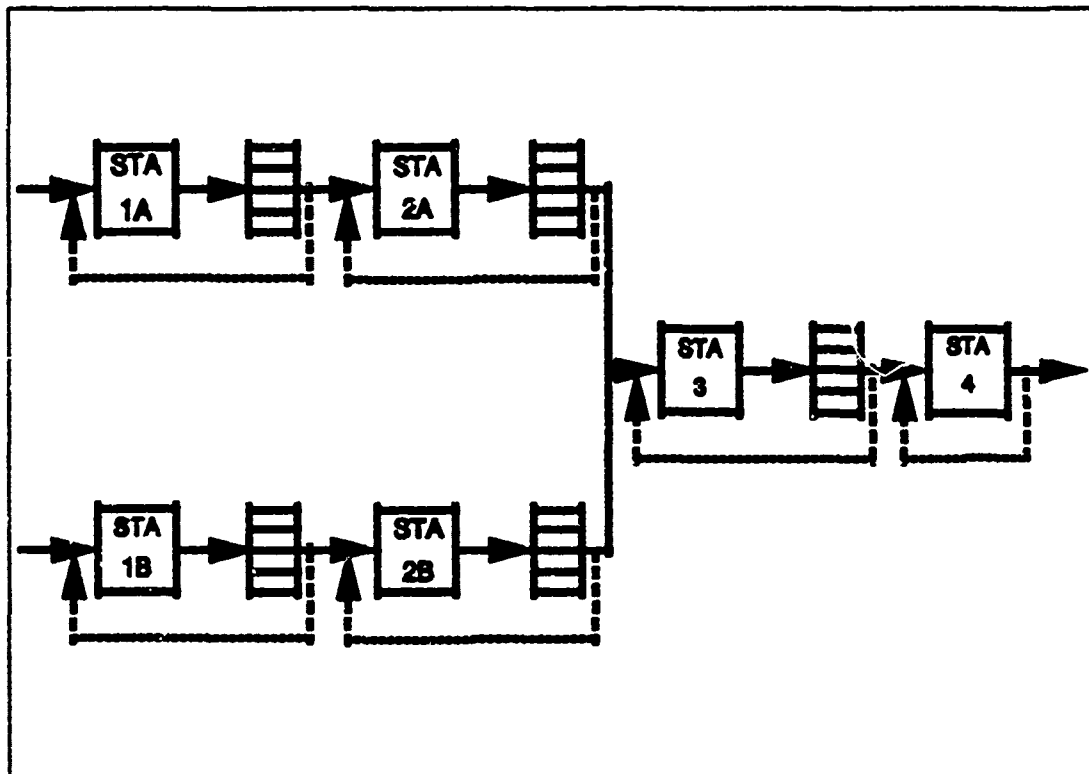
Figure 3-7. System 2 - JIT with Kanban

The programming for System 2 - Drum-Buffer-Rope was
similar to the flow that was programmed in the previous DBR
model with the exception of the parallel processes. The
work-in-process was calculated as in the previous JIT model
and Figure 3-8 illustrates the flow of this model. The flow
through the model was established by splitting the input
product into two separate entities that were rejoined
immediately before the constraint operation. The control
released both parts of the product into the model at the
same time as the buffer before the constraint requires
additional inputs.
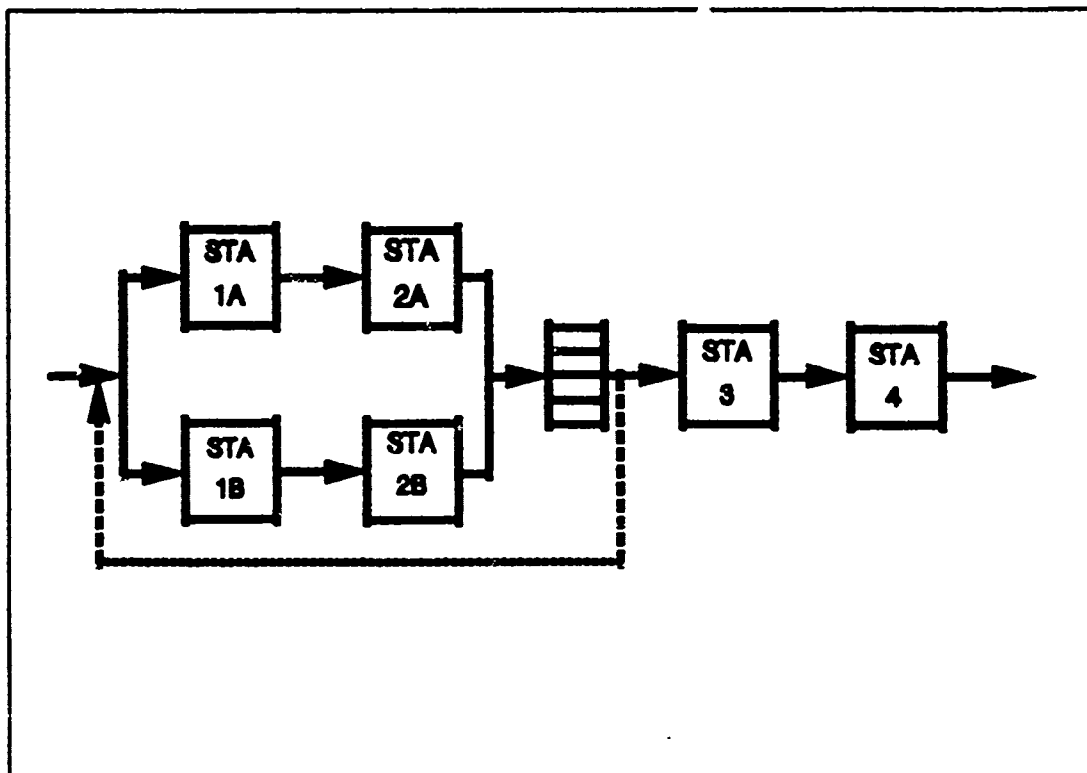
Figure 3-8. System 2 - Drum-Buffer-Rope

The programming for System 3 - JIT with kanban was similar to that established in the previous JIT models with the exception of the production of two products. The work-in-process considers the subcomponents in the initial processing stations, Stations 1 and 2, as half of a full unit; while the product in Stations 3 and 4 were considered as full units. The throughput data output from this model included the separate data on Product 1 and 2, as well as, the total throughput for the system. The programming followed the flow diagram of Figure 3-9.
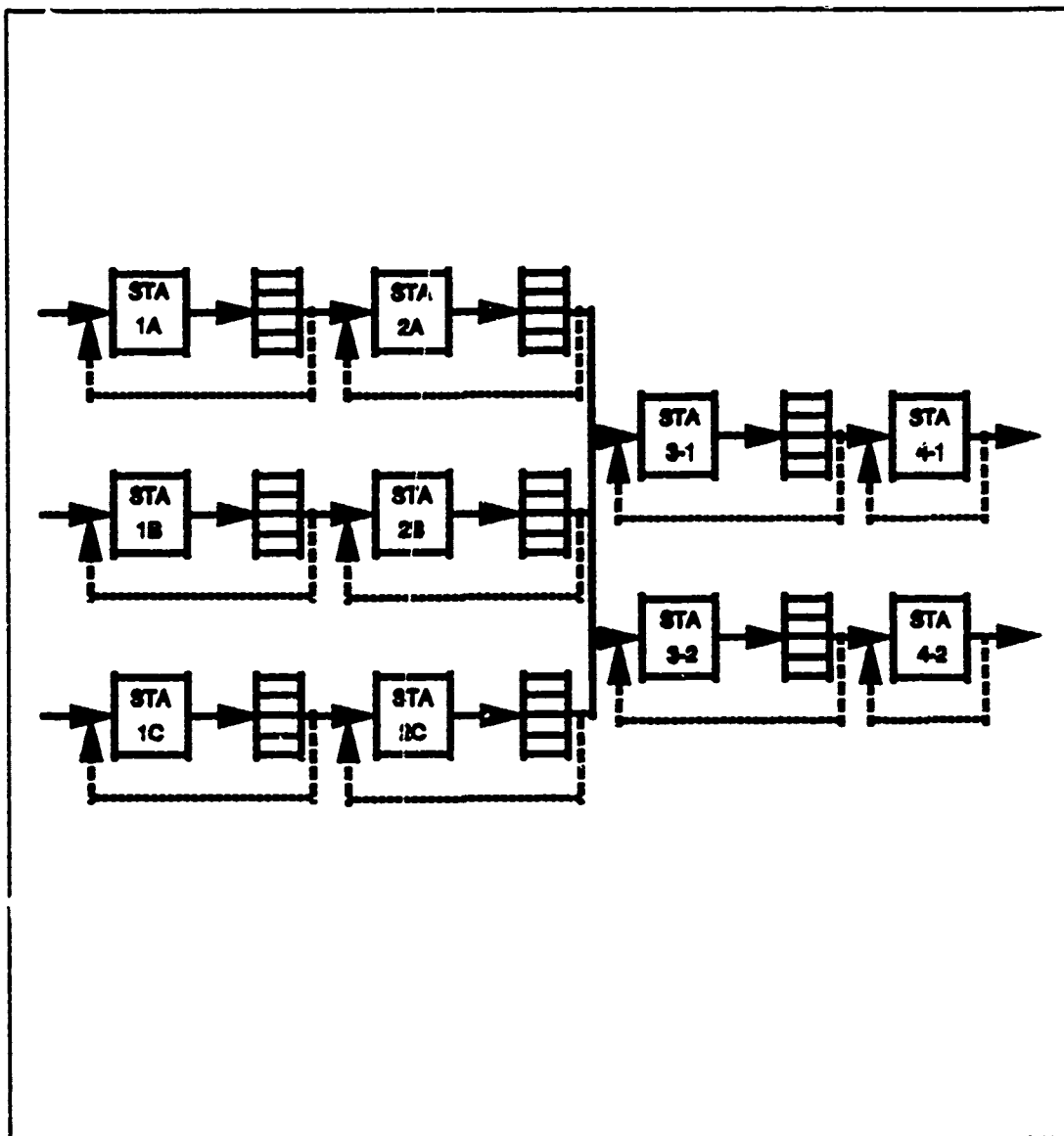
Figure 3-9. System 3 - JIT with Kanban

The programming of System 3 - Drum-Buffer-Rope consisted of all requirements of the previous DBR models and the previous System 3 - JIT model. The programming was developed using the product and flow diagram illustrated in Figure 3-10.
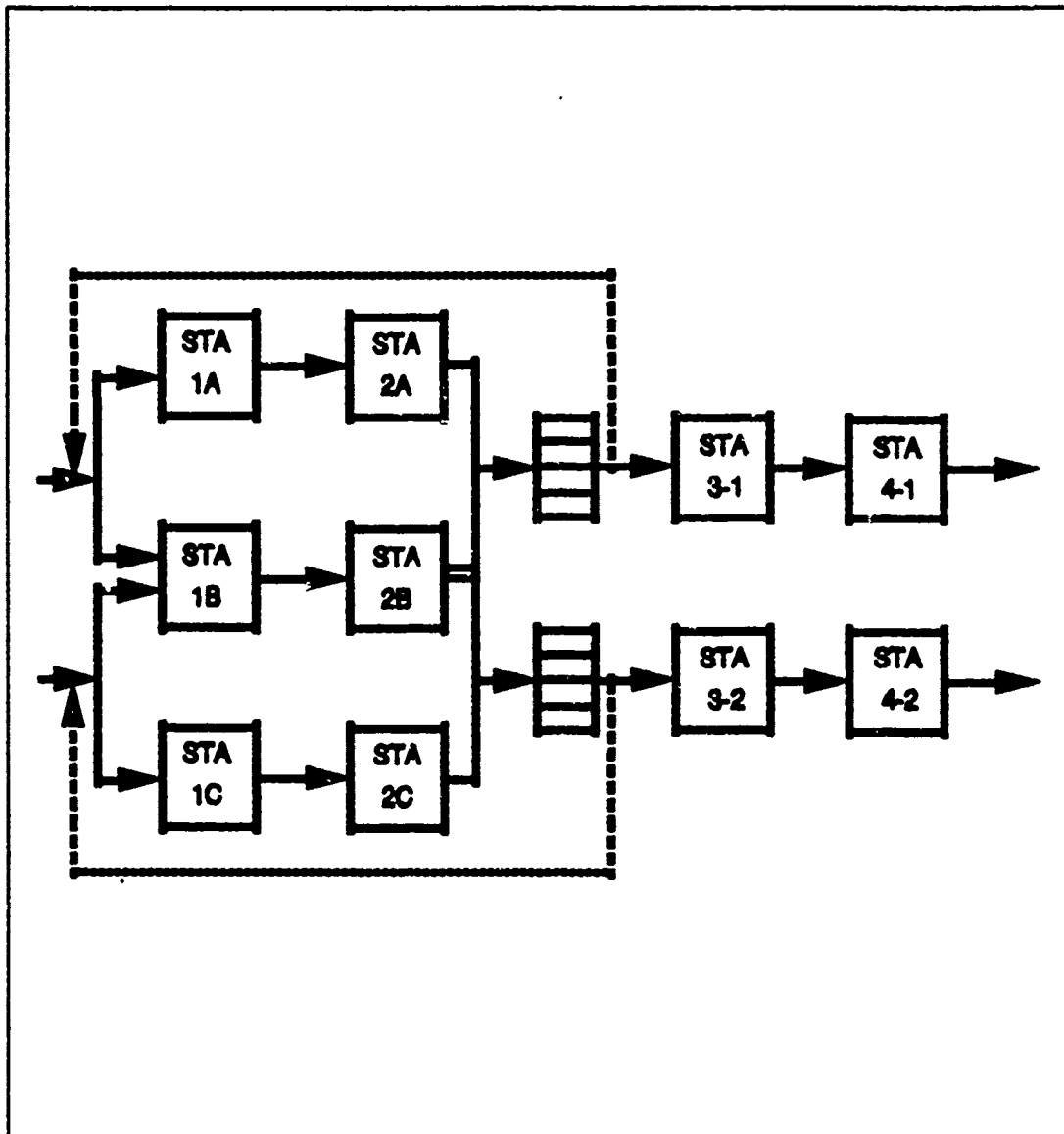
Figure 3-10. System 3 - Drum-Buffer-Rope

Processing Times

System 1. The mean times for each of the process stations of System 1 are listed in Table 3-1. As can be seen from the values, Station 3 with a mean of 20 minutes is the system constraint for both models. The times

for each station will be identical for the models of both scheduling methodologies.

TABLE 3-1

SYSTEM 1 - MEAN PROCESSING TIMES

Process Station	Mean Time (Minutes)
1	7.5
2	12.5
3	20.0
4	5.0

System 2. The mean times for each of the process stations of System 2 are listed in Table 3-2. As was the case in System 1, Station 3 is still the system constraint. The addition of the processing times for Stations 1B and 2B are the only differences from the System 1 processing times.

TABLE 3-2

SYSTEM 2 - MEAN PROCESSING TIMES

Process Station	Mean Time (Minutes)
1A	7.5
1B	5.0
2A	12.5
2B	7.5
3	20.0
4	5.0

System 3. The mean times for each of the process stations in System 3 are listed in Table 3-3. There are now two system constraints in the system, Process 3-1 is the constraint for Product 1 and Process 3-2 is the constraint for Product 2. The processing time for Station 1B was reduced from 5.0 minutes to 2.5 minutes to eliminate the possibility of the B parallel branch from becoming a constraint on both products.

TABLE 3-3

SYSTEM 3 - MEAN PROCESSING TIMES

Process Station	Mean Time (Minutes)
1A	7.5
1B	2.5
1C	5.0
2A	12.5
2B	7.5
2C	10.0
3-1	20.0
3-2	25.0
4-1	5.0
4-2	5.0

Independent Variables. The variables that were studied using the model systems are:

1. Type of scheduling system utilized.

2. Maximum work-in-process inventory present in the system during the simulation run. The maximum work-in-process level for the simulation run was set by designating the size of the buffers in the model. The size of the buffers in the system ranged from 1 unit to 8 units.

3. Probability of failure of process station. Nine different probability values, ranging from 0 to 0.4 in 0.05 increments, were established for the simulation testing. The probability of failure for each station in the model was the same for the simulation run with the exception of the system constraint which did not have a probability of failure.

4. Variance of processing time for process station of the system. Nine different variances, ranging

from 0 to 40 percent in 5 percent increments of the mean process time, were established for the simulation testing. The percentage variability will be plus or minus the percent variability so that a station with a mean processing time of 20-minutes varying at 40 percent encountered processing times ranging from 12-minutes to 28-minutes. All stations within the model, including the system constraint, will have the same percent variability.

Dependent Variables. As has been stated by Dr. Goldratt, one of the measures of a production system is throughput, so the amount of product generated in a set amount of time was the measurement variable. Throughput is also easily measured in the simulation environment.

Data Generation. The probability of process station failure and variation of process time were each evaluated separately using the same testing hierarchy.

Testing Hierarchy. Each of the model systems was scheduled using JIT and TOC methodologies. The model execution sequence consisted of a simulation run, a set-point group, an equi-inventory group, an equi-variability group, and model execution.

1. Simulation Run - consists of an initial 40-hour warm-up period to load all system buffers, followed by an 160-hour data gathering period. The simulation run is a single execution of the particular model and is therefore the smallest unit of the study. During the simulation run all independent variables were set at a constant value. The results of the simulation run consisted of the values of the independent variables and the number of units completed during the data gathering period, dependent variable.

2. Set-point Group - consists of 100 simulation runs. The set-point group is a collection of simulation runs each with the same values for the independent variables. Each of the simulation runs within this group utilized different groups of random numbers by starting the particular simulation run at a different point on the random number stream.

3. Equi-Inventory Group - consists of 9 set-point groups each having the same value of inventory. but each set-point group had different values of variability.

4. Equi-Variability Group - consists of 8 set-point groups each having the same variability value, but different inventory. The equi-variability group was either a process time equi-variability group or a station failure equi-variability group. Each of the set-point groups in the utilized the same random numbers.

5. Model Execution - consists of 72 set-point groups with all combinations of inventory and variability tested. The two types of model execution are process time variation and station failure. Both of these model executions were conducted using both JIT and TOC scheduling on all three of the production systems. There were a total of 12 model executions in this study.

Data Set. The data set for a model execution consisted of 7,200 lines of data. Each data line consists of the output data for the particular simulation run.

Data Acceptability. Even though the systems that are being simulated are purely theoretical, the model executions were evaluated to insure that they were performing as expected. To insure that the data generated by the model execution is as expected the following steps were taken.

1. To insure that each random process in the model encounters a statistically random set of random numbers each process was driven off of a different and unique random number stream. This insured that no process encountered either consistently low or high random numbers during a simulation run.

2. The results of the JIT model executions were compared against the results achieved in previous JIT performance evaluation efforts (Lulu, 1986; Lulu and Black, 1987).

Data Analysis. A point estimate for the throughput mean and standard error of the mean was calculated using the

data of each set-point group. The point estimate consists
of the calculation of the average throughput for the set-
point group and the sample variance of the throughput.
These values were then utilized to calculate upper and lower
confidence limits. Since the sample size was n=100 then,
through the Central Limit Theorem, it can be assumed that
the mean throughput is normally distributed (Devore,
1987:212-214). Since the mean throughput is approximately
normally distributed the confidence limits were calculated
using the standard normal distribution. Confidence limits
were calculated around each of the estimation points. For
equal levels of work-in-process, equi-inventory group data
for the TOC model was compared to the JIT model for each of
the systems and both types of variability using graphical
comparison. The confidence limits were be graphed along
with the average throughput for each level of variability to
determine if the mean throughputs are statistically
different. The hypothesis was that the TOC scheduled system
would yield higher throughput than the JIT scheduled system.
If the confidence limits for the mean values do not overlap
then it can be assumed with 95 percent confidence that they
are not equivalent and the scheduling methodology with the
larger average throughput yielded superior performance.

The production systems were compared using only the TOC
scheduling methodology to determine if the flow of the
system affects the resulting throughput. System 1, System
2, and Product 1 of System 3 all utilize essentially the

same longest path processing times so comparison between these throughputs will be made. These comparisons were again made graphically using equi-inventory group data for the equivalent buffer size setting for each of the systems. The hypothesis was that there would be no difference in the throughput due to differing system flow.

Statement of Decision Rules. The Type I Error that was allowed is 0.05 level of significance. This indicates that the level of confidence is 95 percent. If the expected throughput for the JIT scheduled systems are less than this 95 percent confidence level then the hypothesis that the expected throughput of the TOC scheduled system is greater than systems scheduled with other methodologies will not be rejected.

IV. Results and Analysis

Introduction

This research effort was to investigate and compare the effects of system variability on systems scheduled using both Just-in-Time (JIT) and the Theory of Constraints (TOC). This chapter provides the results of the execution of the experimental design developed in the previous chapter. The data generated by the simulation models will be analyzed for both the JIT and TOC models, and then the results will be compared at the lower equivalent work-in-process (WIP) levels. The analysis and comparison will be conducted in accordance with the experimental design outlined in the previous chapter. The experimental design was modified slightly by additional simulation runs of System 1, both JIT and TOC models, for larger variability of processing time. The additional runs were conducted to investigate a downward trend of the JIT model that was noticed at the 40 percent variability level. The analysis will be conducted for each of the three systems separately, then the results of the TOC models for each of the systems will be compared. For each system the effects of process time variability will first be analyzed for both scheduling methodologies. Then the results for the scheduling methodologies will be compared. This sequence will then be repeated for variability of process station failure.

Findings

System 1. Internal to the system models (Appendix A and B) the buffer size was set at a value ranging between 1 and 8; this setting of the buffer size caused the maximum work-in-process (WIP) level for the JIT system to be greater than the level of the TOC system. For example, for a buffer size of 1 the JIT system would be capable of having one unit of product in each of the four process stations so the maximum WIP level is 4. The TOC system can have only 1-unit in Stations 1 and 2, 1-unit in Station 3, and 1-unit in Station 4 for a total of 3-units. So for a buffer size of 1 the JIT system will have a WIP level of 4 while the TOC system has a WIP level of 3. This difference, due to the buffering arrangement of the methodology, can be seen in Table 4-1. Table 4-1 also contains the line legend for the graphs of the System 1 and 2 results.

Process Time Variability. The process time variability value is the percent variability around the mean processing time for the station. For example, for a processing station with a mean of 20 minutes and a variability value of 0.2 or 20 percent, the actual time can range between 16 and 24 minutes. The percent variability ranged from 0 to 40 for the model executions. A baseline throughput value was generated by running a model with no variability. For both of the model executions the baseline throughput value was 480 units, which was the expected value for the system.

TABLE 4-1

WORK-IN-PROCESS LEVELS AND GRAPH LEGEND - SYSTEMS 1 AND 2

Buffer Size	JIT WIP Level	TOC WIP Level	Graph Line
1	4	3	————————
2	6	4	————————
3	8	5	————————
4	10	6	————————
5	12	7	————————
6	14	8	————————
7	16	9	————————
8	18	10	————————

JIT with Kanban. The results of this model
execution were compared with the results of the 1987 study
of the same type by Lulu and Black, and the results of both
this and the Lulu and Black study are essentially identical.
The finding of the Lulu and Black study was that a JIT
system was not significantly affected by process time
variability (Lulu and Black, 1987:215); this same result can
be seen in Figure 4-1. The lines for the different levels

of WIP vary some around the baseline throughput value of 480
units, but there are no significant deviations from the
baseline value. There is a slight drop in the throughput of
the lowest WIP level, buffer size 1, in the 35 to 40 percent
variability range. This drop will be investigated further
in the analysis of the data. The program which generated
the Figure 4-1 data is in Appendix A.
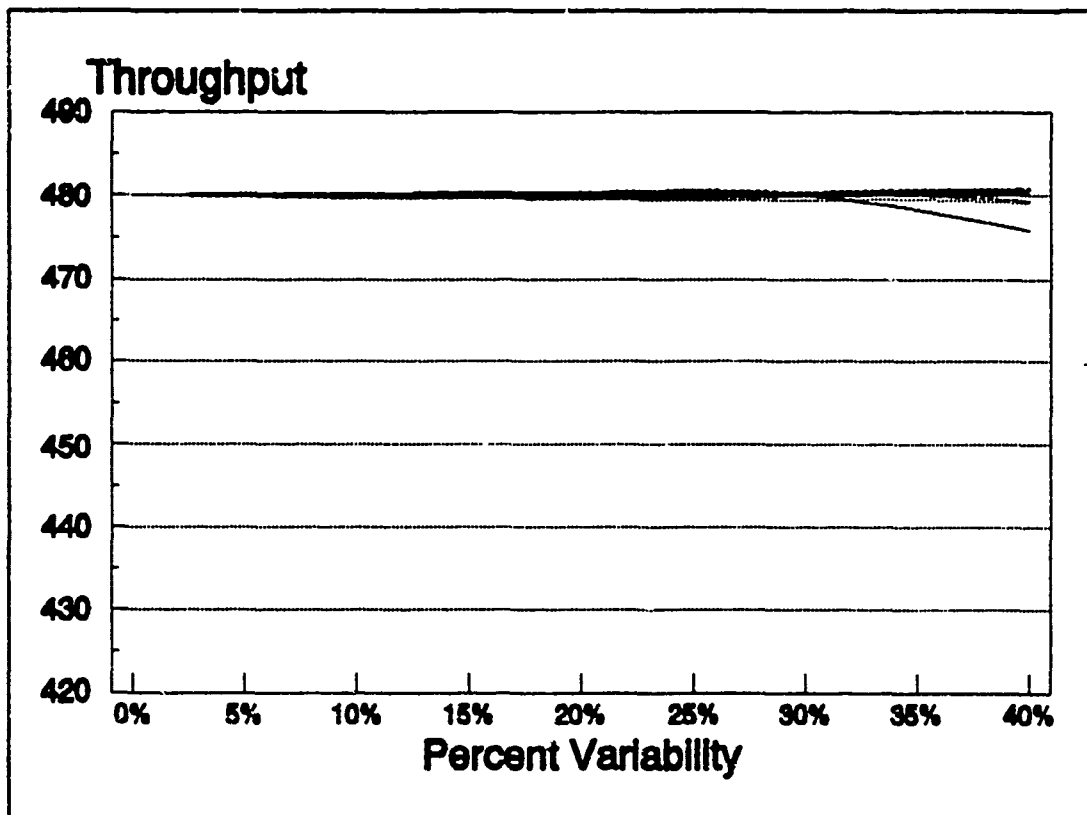
Figure 4-1. System 1 - JIT - Process Time Variability

TOC - Drum-Buffer-Rope (DBR). There are no
previous published studies which evaluate DBR systems to
provide a comparative baseline for the results. The only
level that deviates from the results of the JIT model
execution is the lowest WIP level which drops sharply below
the baseline value (Figure 4-2). This result was achieved
at a WIP level of 3-units, buffer size 1, and is due to the
starvation of units to be processed by the constraint. The
buffering of the DBR control methodology caused the initial
two processing stations to, in effect, become one station
equal in processing time to the constraint. The
specification of a buffer size of 1-unit stipulated that
only one unit could be between the material release point
and the constraint. As the variability of the processing
time increased the throughput of the system was disrupted
since the system constraint was not fixed. The constraint
could be either the normal constraint, Station 3, or the
constraint created by the combination of Stations 1 and 2
depending on the actual processing times for the stations.
The combination of Stations 1 and 2 into a constraint
created a situation where the buffer for the normal
constraint worked against achieving the baseline throughput.
This situation illustrates the importance of properly sizing
the system buffer and reducing the variability of the system
which are both part of the TOC philosophy. The program
which generated the data for Figure 4-2 is in Appendix B.
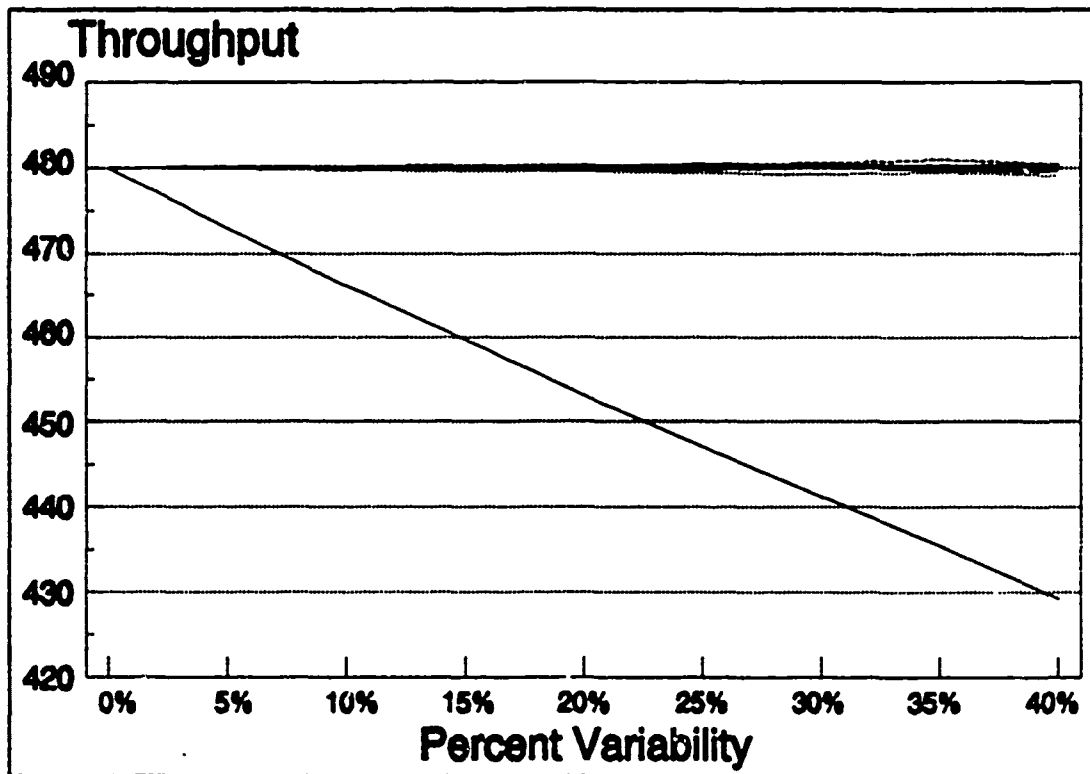
Figure 4-2.   System 1 - TOC - Process Time Variability

Comparison of JIT and TOC.   The comparison of

the two scheduling systems was conducted at the lowest two

levels of work-in-process (4 and 6-units) for the JIT model.

To avoid a confound, equivalent inventory levels were used

for comparison.   Since the philosophy of both methodologies

emphasizes the minimization of WIF in the system the

comparison at higher levels is not consistent with the

philosophy.   It can also be seen from the previous graphs

(Figures 4-1 and 4-2) that for high levels of WIP, both

methodologies produce the baseline throughput for all levels
of variability.

At the 4-unit level it can be seen from the graph
(Figure 4-3) that, at process time variability of less than
35 percent, the two scheduling methodologies produce similar
results that are not statistically different. At levels of
variability greater than 35 percent the TOC system, large
solid line, produces throughput that is higher than the JIT
system, large dotted line, and the difference is
statistically significant ($\alpha = 0.05$) since the confidence
intervals do not overlap.
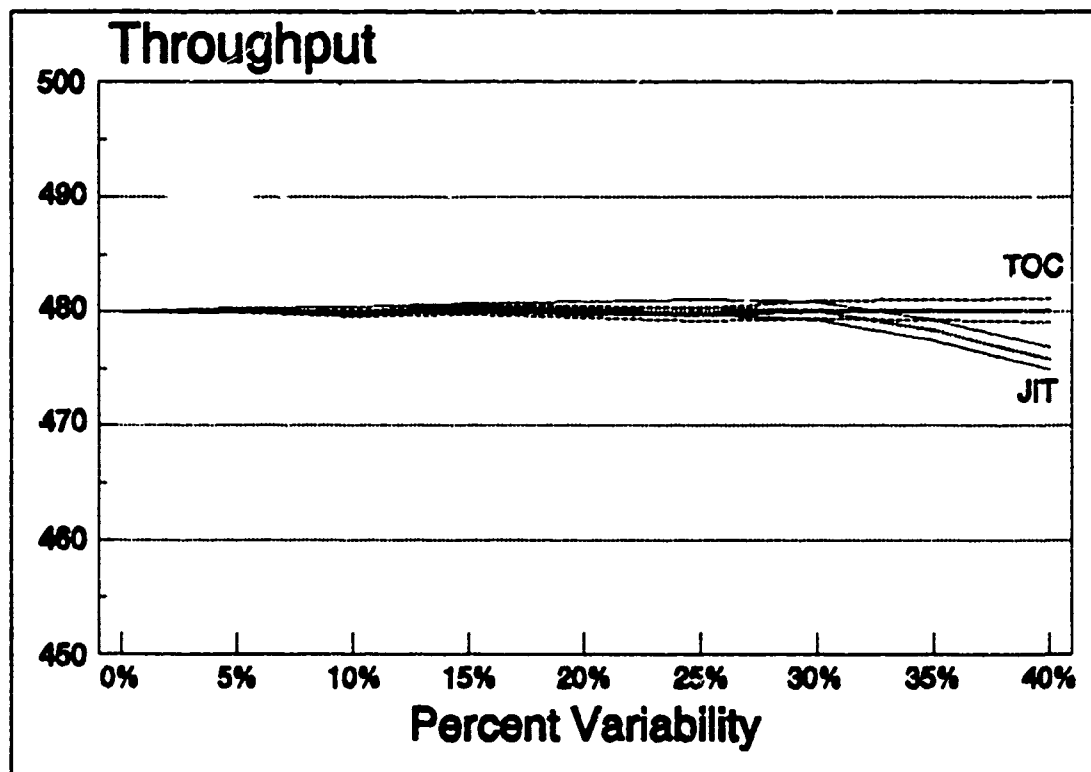
Figure 4-3. System 1 - Comparison at WIP = 4-units -
Process Time Variability

It can also be seen that while the TOC throughput
remains at essentially the baseline value at the 40 percent
variability level the JIT throughput begins to diverge.
These systems were run again with the WIP inventory set at 4
units and the variability ranging from 0 to 75 percent. The
results of this additional equi-inventory group run can be
seen in Figure 4-4.
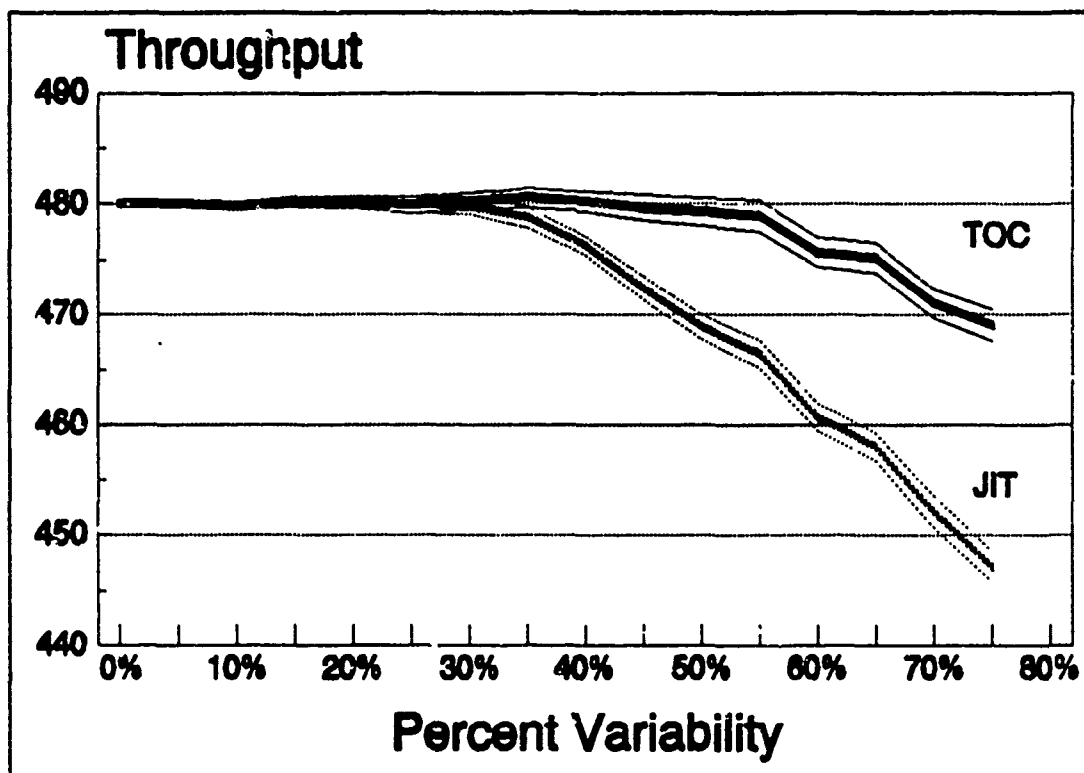
Figure 4-4.   System 1 - WIP = 4-units - High Process
              Time Variability

It can be seen from this graph that at very high levels
of variability the TOC system significantly outperforms the
JIT system. The deleterious impact of variability is

4-8

acknowledged by the variability reduction techniques which
are inherent in both philosophies.

The comparison between JIT and TOC at 6 units of WIP
was conducted only over the variability range from 0 to 40
percent. It can be seen from the graph (Figure 4-5) that
there is no statistical difference between the two systems
at this level of WIP inventory. The oscillations around the
baseline level of throughput are normal experimental error
as the baseline value is always within the confidence
interval for both systems.
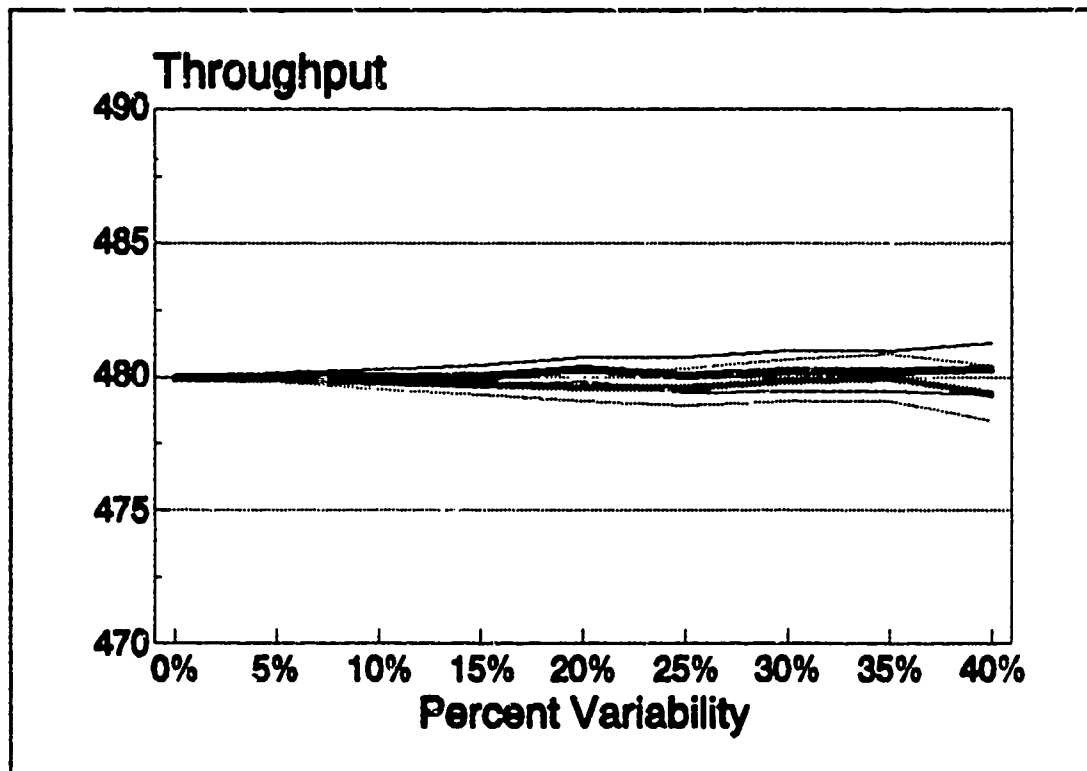
Figure 4-5. System 1 - Comparison at WIP = 6-units -
Process Time Variability

Probability of Station Failure. The probability
of failure value is the probability that the process station
will experience a failure when a unit attempts to enter. As
in the previous tests, the baseline throughput value was 480
units.

The line legend for all of the graphs of probability of
station failure is in Table 4-1.

JIT with Kanban. The throughput of the JIT
system was degraded as the probability of station failure
increased (Figure 4-6). This is consistent with results
documented by Lulu (1986:237). While the throughput did
increase as the WIP level increased the level did not reach
the baseline level of 480 units in any of the trials for
high probability of failure values. The throughput values
seem to be converging on a single value, so that as even
higher levels of WIP are used the throughput curve does not
change. This is the expected result since as the WIP level
increases the availability of material in the system is no
longer a limiting factor and the time required to return the
stations to full operational capability begins to dictate
the throughput of the system. This type of degradation of
throughput illustrates the importance of the system·
reliability aspects that are inherent in the JIT philosophy.
It can be seen from Figure 4-6 that as long as the
probability of a station failure is small the throughput
remains at the baseline value even for low WIP values.

The program which generated the data for Figure 4-6 is in Appendix A.
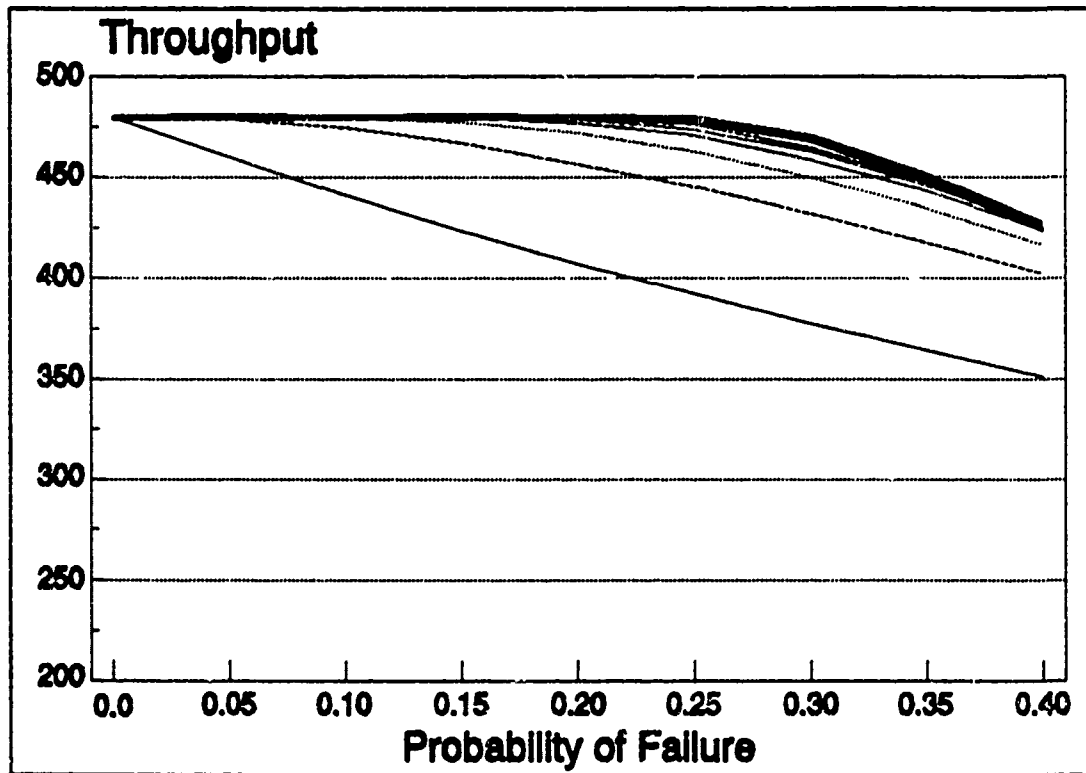
Figure 4-6. System 1 - JIT - Station Failure

TOC - Drum-Buffer-Rope. The results of this model execution are similar to the results achieved by the JIT model. For high probabilities of failure the throughput does not reach the baseline value even for very high levels of WIP (Figure 4-7). The values of the throughput, as in the JIT system, appear to be approaching a constant value for a particular probability of failure.

4-11

The program which generated the data for Figure 4-7 is
in Appendix B.
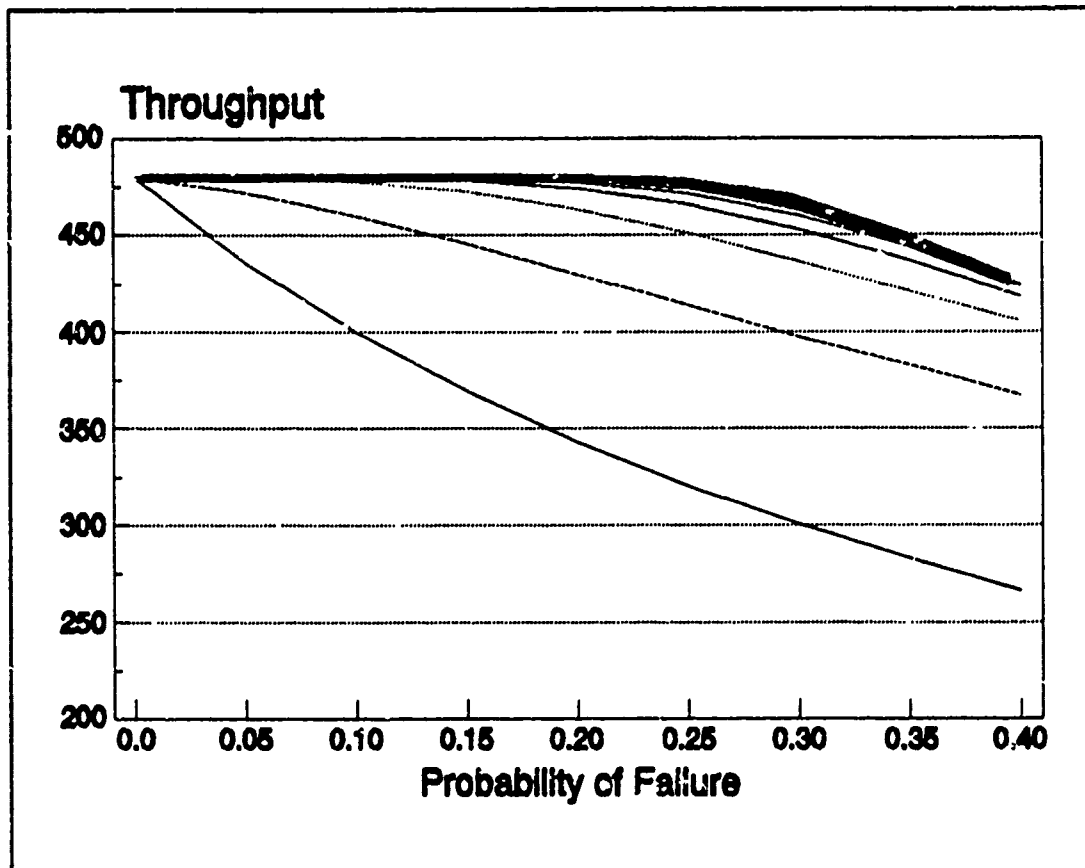
Figure 4-7.   System 1 - TOC - Station Failure

Comparison of JIT and TOC.   The comparison
between the two methodologies was conducted at 4 and 6 units
of WIP and results illustrated in Figures 4-8 and 4-9.   The
JIT throughput curve is the heavy dotted line and the TOC
curve is the heavy solid line.   The confidence interval
lines can not be seen clearly in these graphs due to the
very narrow width of the confidence interval and the large

scale required by the throughput values. The data for Figures 4-8 and 4-9 was generated by the programs in Appendices A and B.

At the 4-unit level the TOC system achieves a higher throughput than the JIT system for all probabilities of failure (Figure 4-8). The difference between the curves is statistically significant at the 95 percent confidence level ($\alpha = 0.05$).
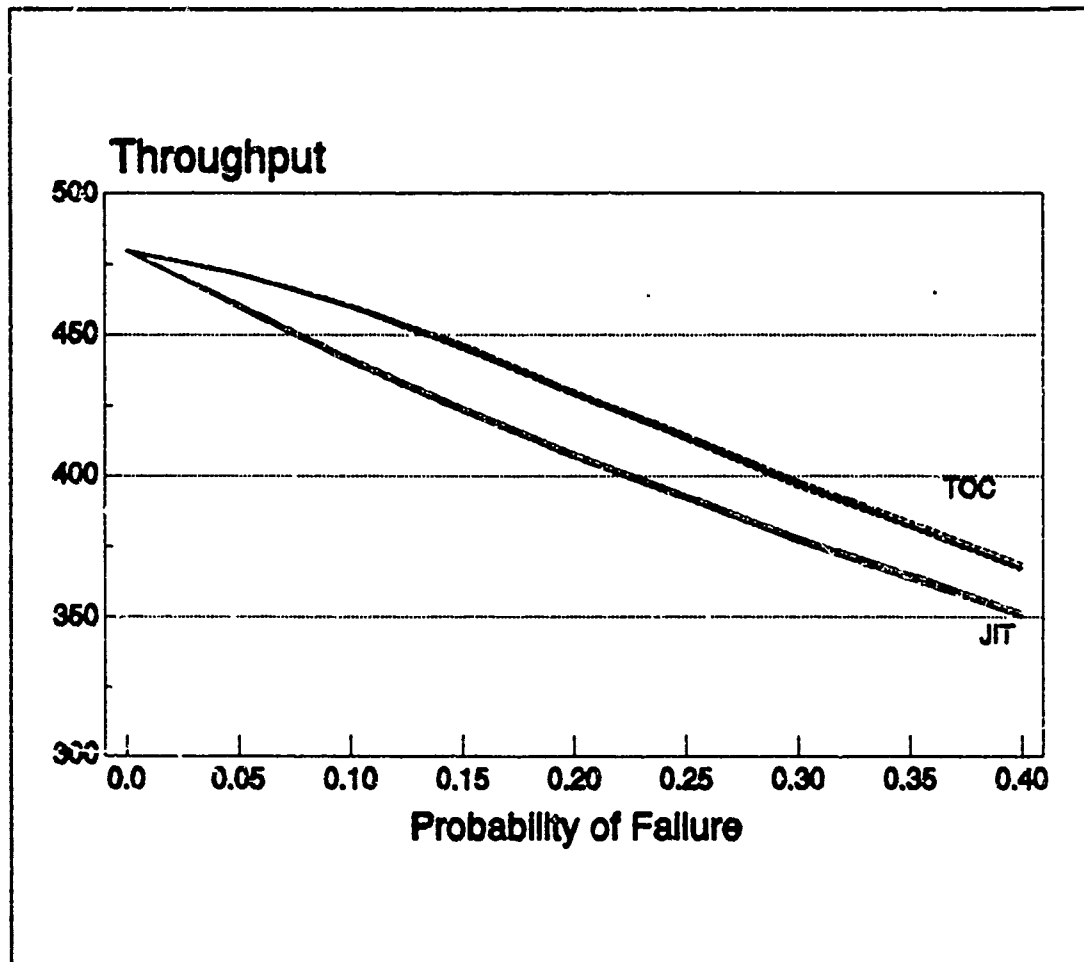
Figure 4-8. System 1 - Comparison at WIP = 4-units - Station Failure

At the 6-unit level of WIP the TOC system again achieves a higher throughput for all probability values of station failure (Figure 4-9). The curves again drop off significantly below the baseline value as the probability of failure increases. Since the confidence intervals for the two curves do not overlap the difference between the scheduling methodologies is statistically significant ($\alpha = 0.05$).
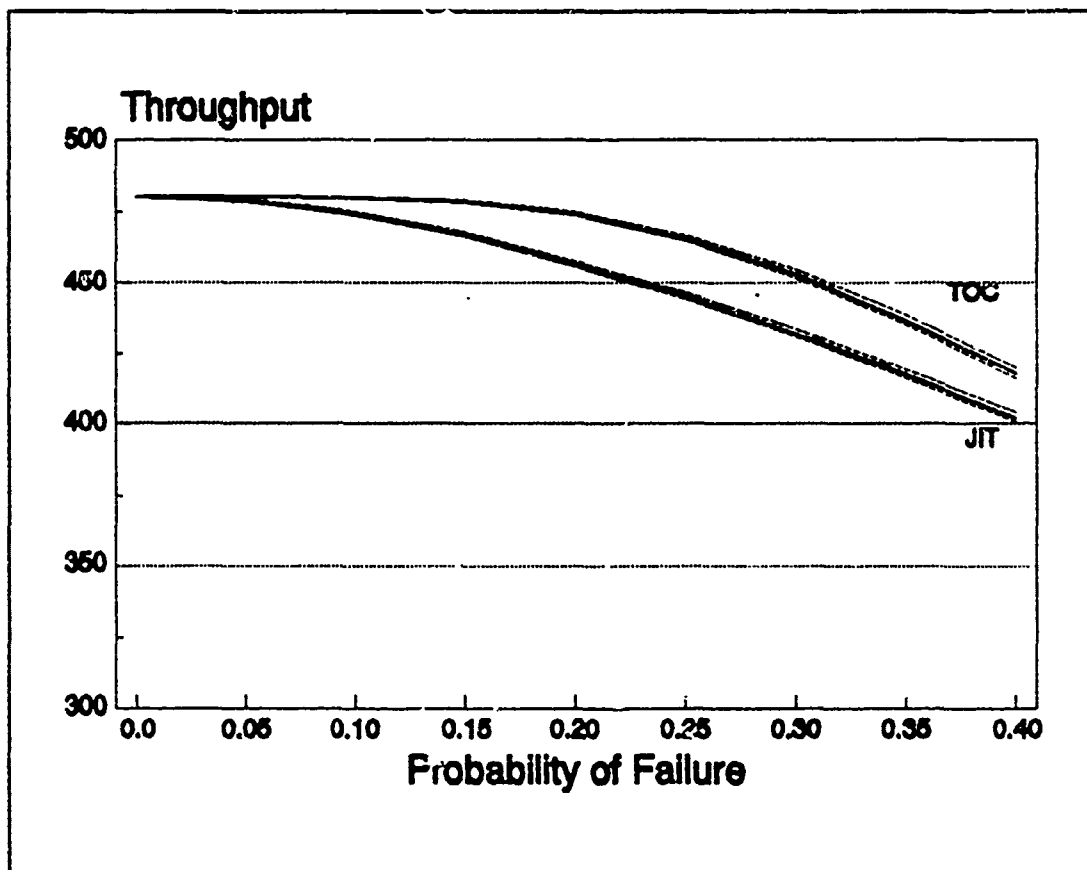
Figure 4-9. System 1 - Comparison at WIP = 6-units - Station Failure

System 2. The results from the model executions for System 2 were essentially identical to the results obtained from the System 1 model executions. The buffer sizing and graph line legend is the same as in System 1 so it will not be repeated here. The WIP level for the JIT system again ranged from 4 units (buffer size = 1) to 18 units (buffer size = 8) in increments of 2 units; the TOC system's WIP ranged from 3 units to 10 units in increments of 1 unit.

Process Time Variability

JIT with Kanban.

The results of this model execution appear in Figure 4-10. The shape of the curve and the values obtained in the point estimates are essentially the same as that of the System 1 execution. The program which generated this data is in Appendix C.

It can be seen in Figure 4-10 that the throughput is essentially constant and not dependent on process time variability. The only exception is at the WIP level of 4-units where the throughput begins to drop at very high levels of variability. This drop was investigated previously in System 1 so it will not be repeated for System 2, but the results are expected to be similar. The remaining WIP levels only vary about the baseline value and do not appear to be affected by the variability.
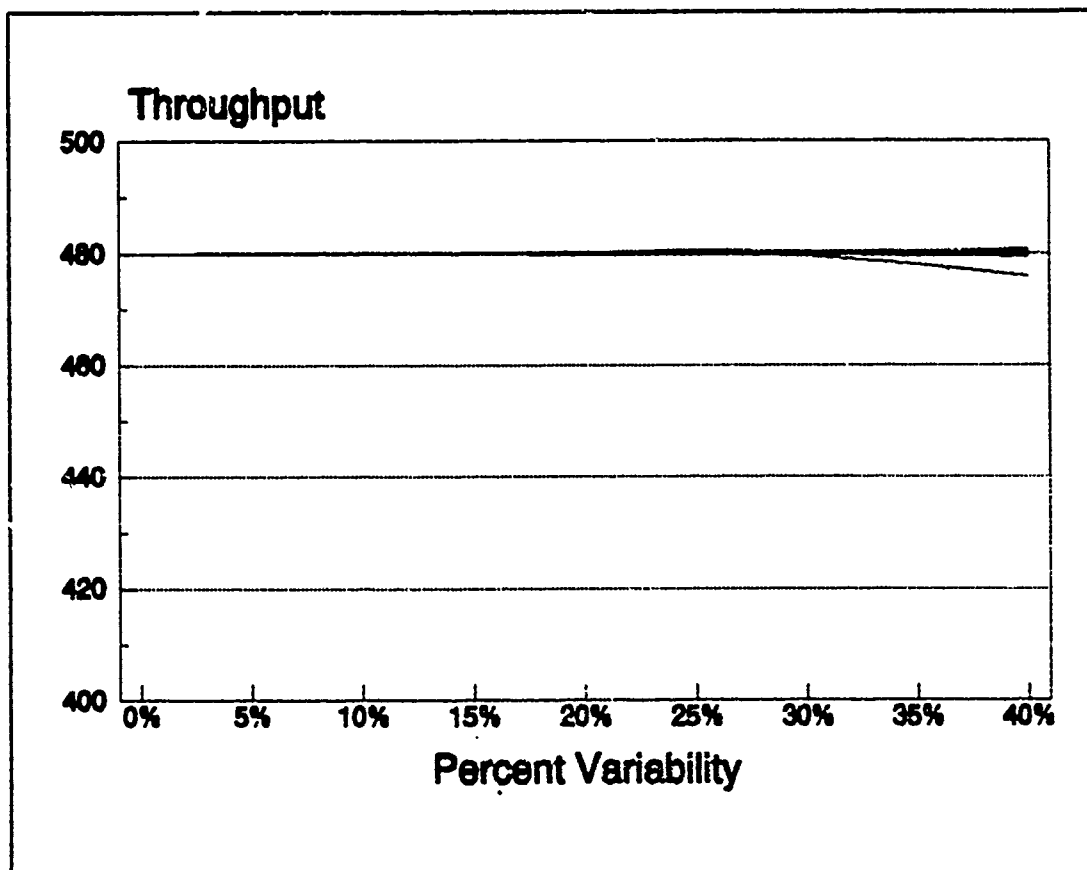
Figure 4-10.   System 2 - JIT - Process Time Variability

TOC - Drum-Buffer-Rope.   The only significant feature of the throughput graph (Figure 4-11) for this model execution is the drop in throughput for the lowest WIP level.   This drop was discussed above in System 1 and there is no change for System 2.   The remaining WIP levels maintain the baseline throughput for all levels of variability.   The program which generated this data is in Appendix D.
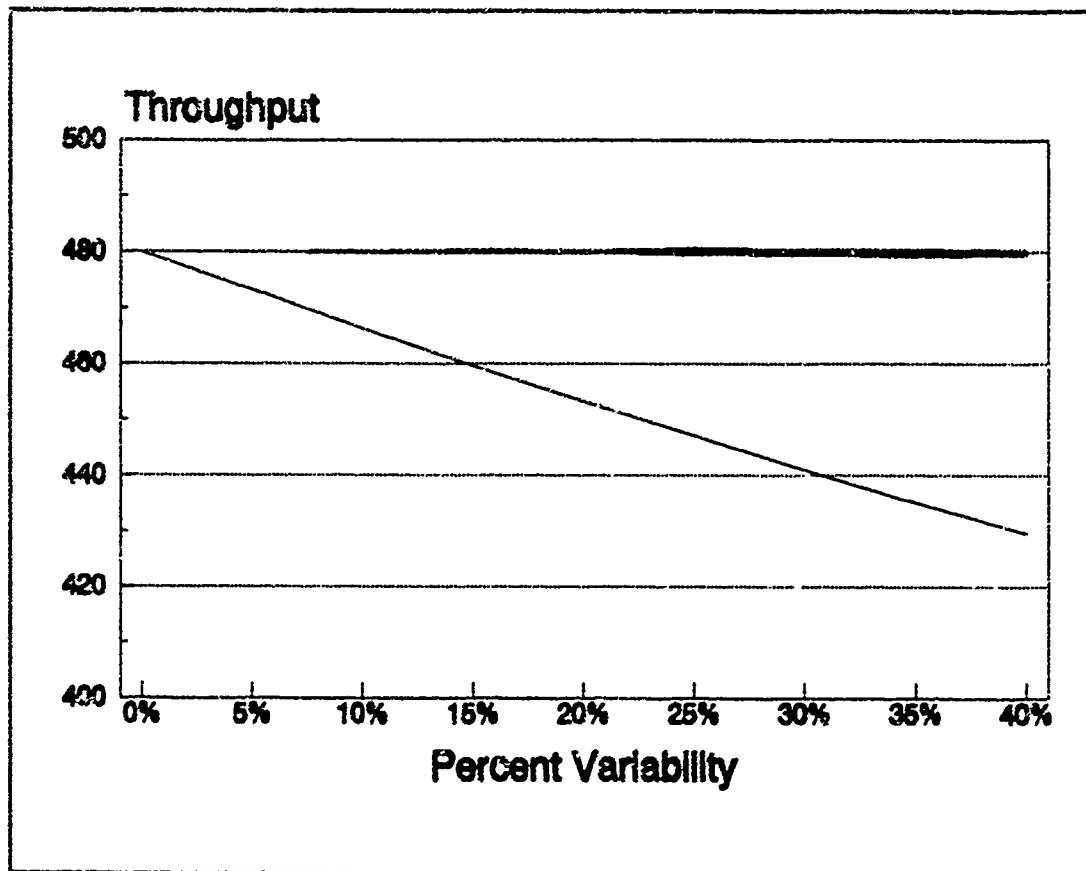
Figure 4-11. System 2 - TOC - Process Time Variability

Comparison of JIT and TOC. The results of
this comparison are the same as System 1 so only the 4-unit
level of WIP will be analyzed. The throughput of the TOC
system (solid line) maintains the baseline value for all
levels of variability (Figure 4-12). The JIT system, dotted
line, begins to drop from the baseline value at the 35
percent variability level and by the 40 percent level the
difference between the two systems is statistically
significant ($\alpha = 0.05$).
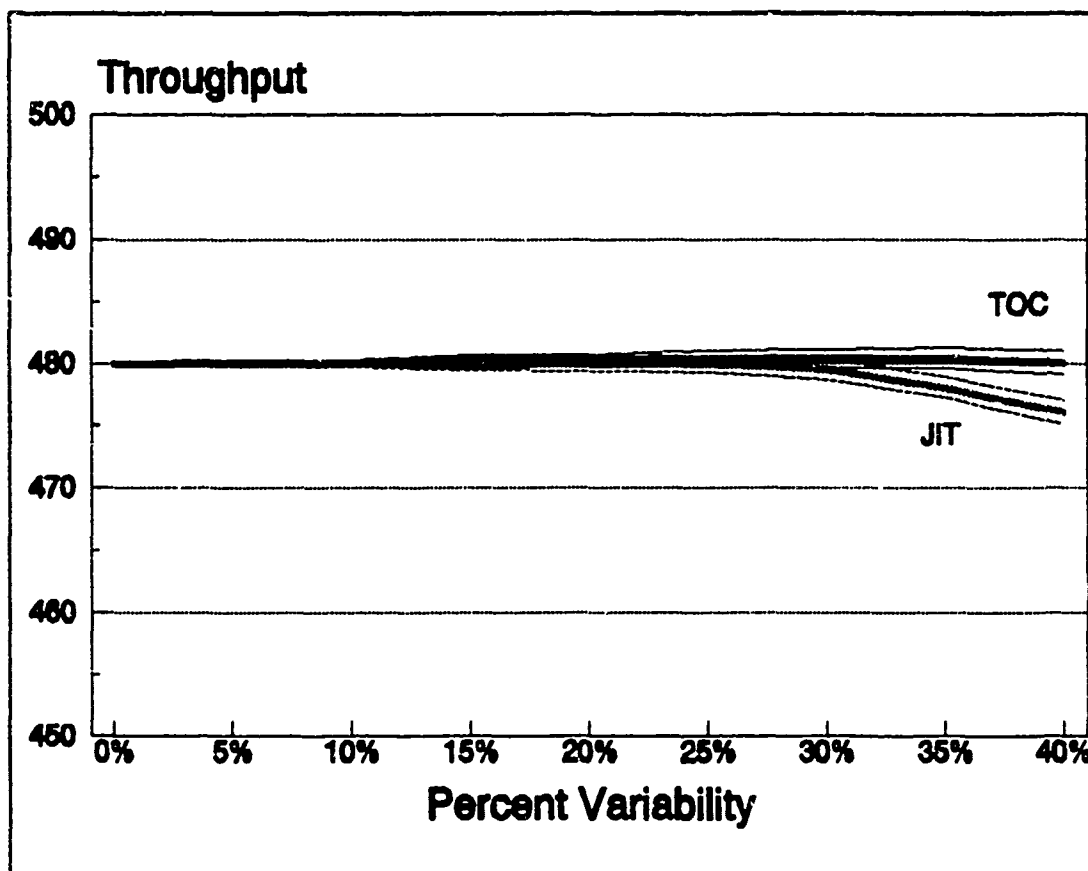
Figure 4-12.   System 2 – Comparison at WIP = 4-units –
Process Time Variability

Probability of Station Failure

JIT with Kanban. The results of this model
execution are essentially the same as was obtained in System
1. If additional information is required the model which
generated the data is in Appendix C.

None of the WIP levels provided complete protection for
throughput and so there was some degradation as the
probability of failure increased. It can be seen in Figure
4-13 that the lower the level of WIP the greater the

reduction in throughput. This degradation is very similar

to that seen in the previous System 1 analysis for the same

levels of failure probability. For low levels of WIP the

JIT system essentially shuts down when a failure occurs and

there is no product to move through the processing station

to the next buffer. For high levels of WIP the repair times

for the stations begin to control the amount of possible

throughput.
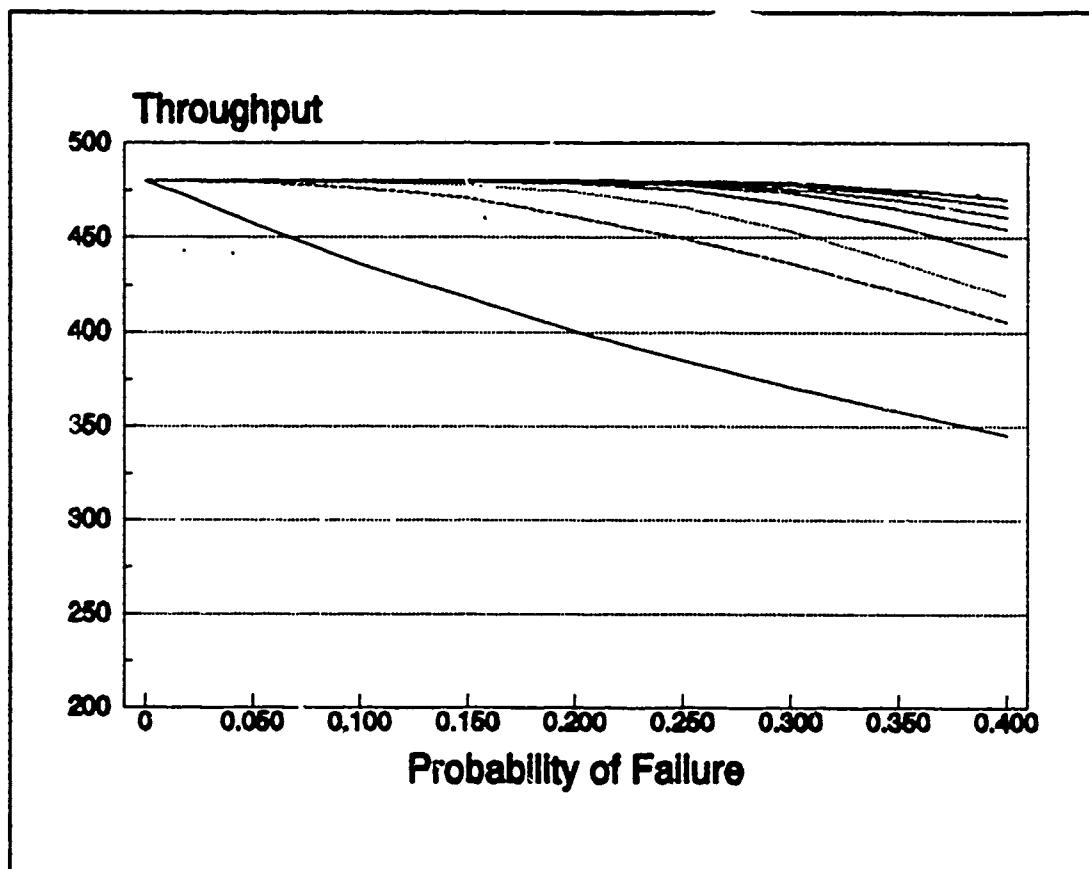
Figure 4-13. System 2 - JIT - Station Failure

TOC – Drum-Buffer-Rope. The results of this model are similar to the previous process station failure models. For all levels of WIP the throughput begins to degrade as the probability of failure increases (Figure 4-14). As in the System 1 analysis, this is the expected result due to the starvation of the constraint caused by station failure in the initial processing stations and the increased flow time caused by the inclusion of station repair time.
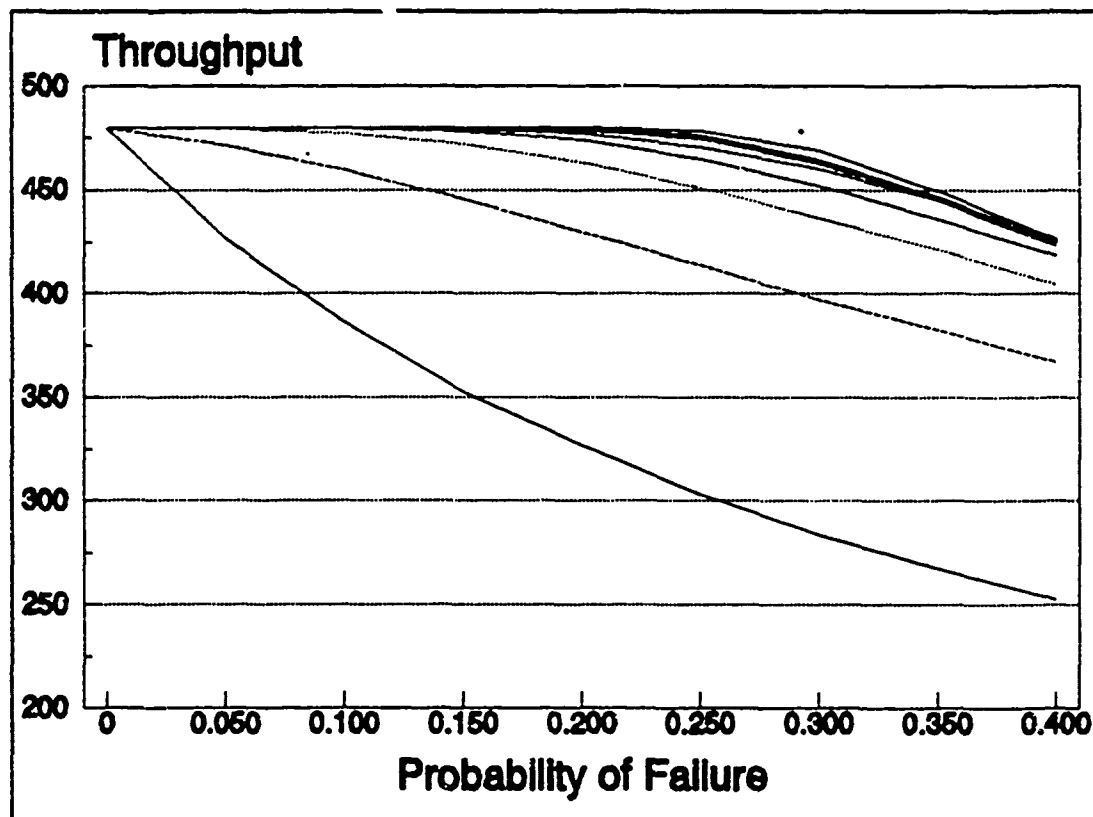
Figure 4-14. System 2 – TOC – Station Failure

Comparison of JIT and TOC. The results of
this comparison and the System 1 analysis are similar as can
be seen by a comparison of Figures 4-8 and 4-15. Due to the
similarity, this comparison will be accomplished only at the
4-units of WIP level. The results of a comparison at the 6-
unit level are expected to be similar to those obtained for
System 1.

At the 4-unit level of WIP the results are, as
expected, closely parallel to those of System 1. The TOC
scheduled system significantly outperforms the JIT system.
The difference between the results is statistically
significant at the $\alpha = 0.05$ level.
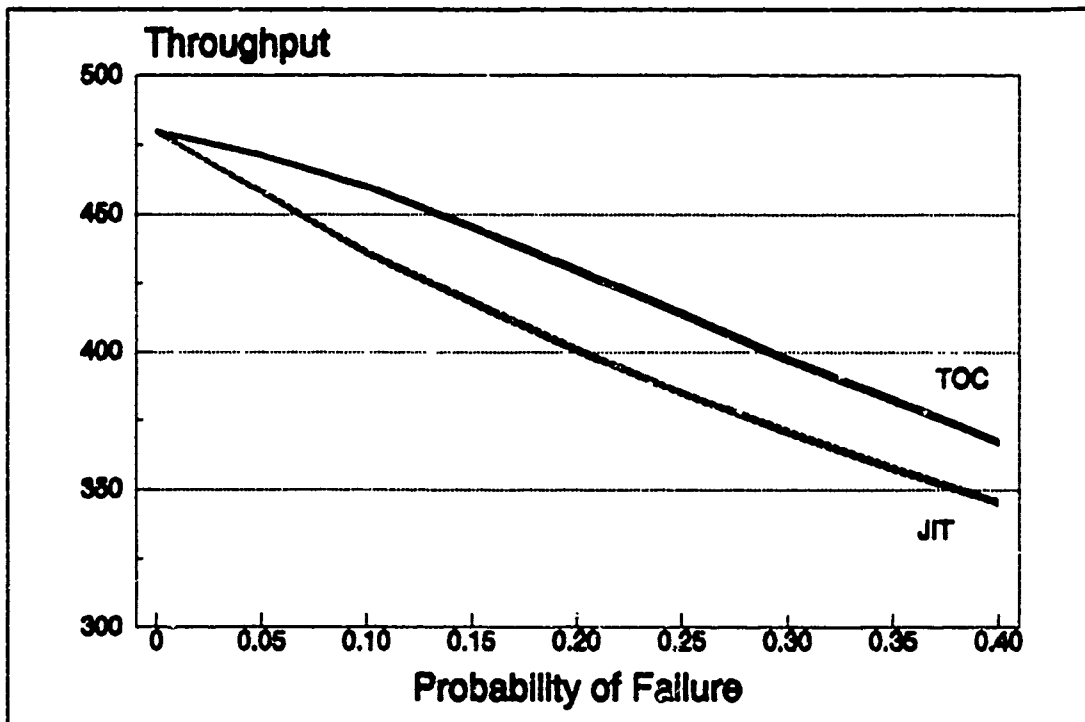
Figure 4-15.   System 2 - Comparison at WIP = 4-units -
Station Failure

System 3. The analysis of System 3 reviews not only the total throughput of the system, but also the throughput of each of the individual products. The results of the System 3 model executions is similar to those seen in the previous systems, but there are differences brought about by the interaction of the two products produced by the system. The buffer size and graph line legend is in Table 4-2 and is applicable to all System 3 figures.

TABLE 4-2

WORK-IN-PROCESS LEVELS AND GRAPH LEGEND - SYSTEM 3

Buffer Size	JIT WIP Level	TOC WIP Level	Graph Line
1	7	6	————————
2	10	8	————————
3	13	10	————————
4	16	12	————————
5	19	14	————————
6	22	16	————————
7	25	18	————————
8	28	20	————————

Process Time Variability. The effects of process time variability on this system are analyzed from the perspective of both the total throughput of the system and individual products produced by the system.

JIT with Kanban. The result of this model execution show a degradation of the total throughput of the system that was not experienced in the previous systems (Figure 4-16). For the lowest WIP level (7-units) the throughput decreased from the baseline value of 864-units to 849-units at 40 percent variability. The program which generated the data is in Appendix E.
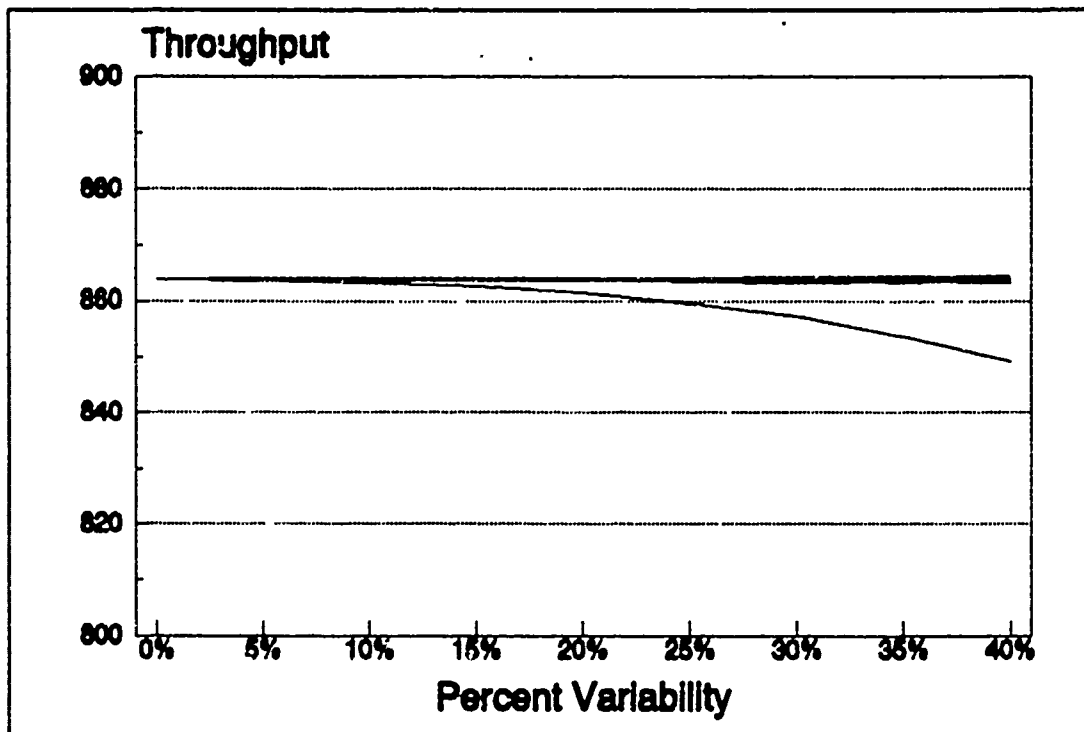
Figure 4-16.   System 3 - JIT - Total Throughput -
Process Time Variability

The degradation in the total throughput of the system can be attributed to the degradation experienced by Product 1. Figure 4-17 shows that the throughput of Product 1 decreased as the amount of variability increased, while Figure 4-18 shows that there was no decrease in the throughput of Product 2. The throughput of Product 2 also does not exhibit the decrease in throughput that was experienced in the previous systems in the 35 to 40 percent variability range.
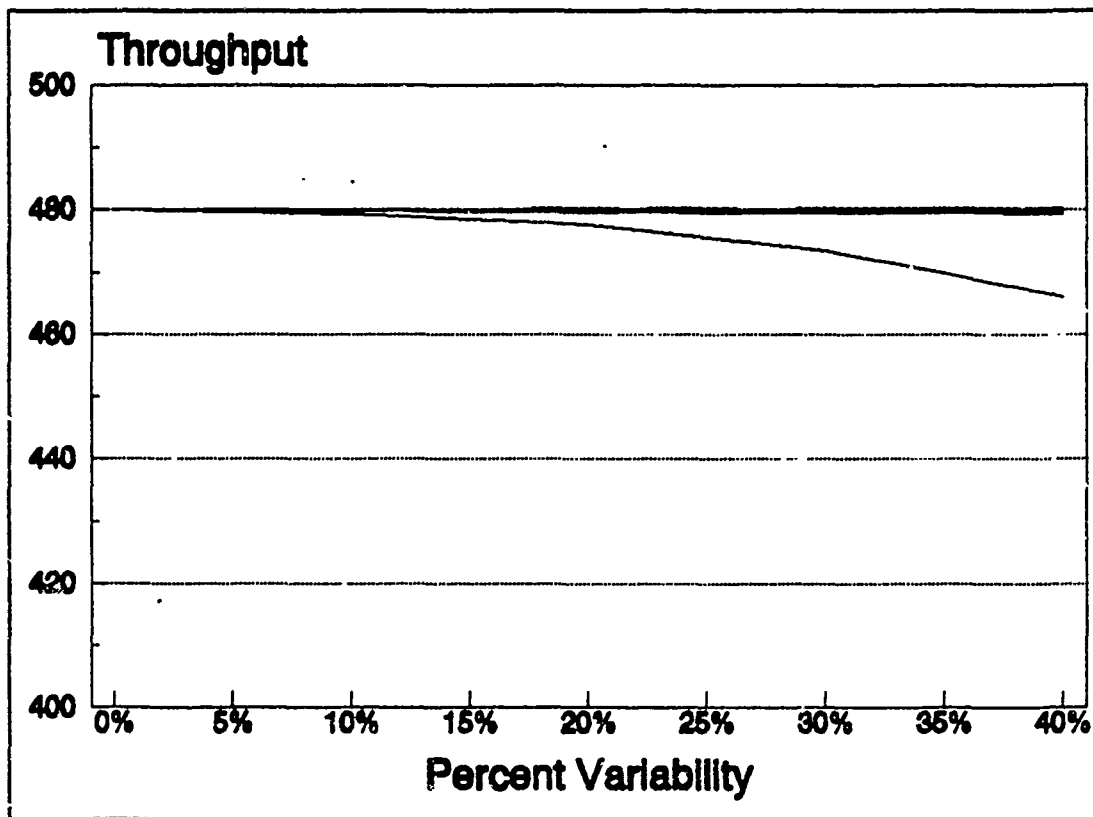
Figure 4-17. System 3 - JIT - Product 1 Throughput -
Process Time Variability

Product 2 may have been able to maintain the baseline throughput value due to the lower processing times in the initial stations for the unique subcomponent in Product 2. Since the unique subcomponent finished initial processing faster than the unique subcomponent for Product 1, Product 2 was able to consume the common subcomponent before Produc⌐ 1. This research effort did not investigate the cause of this finding to determine the exact cause of the irregularity since it was not pertinent to the comparative study.
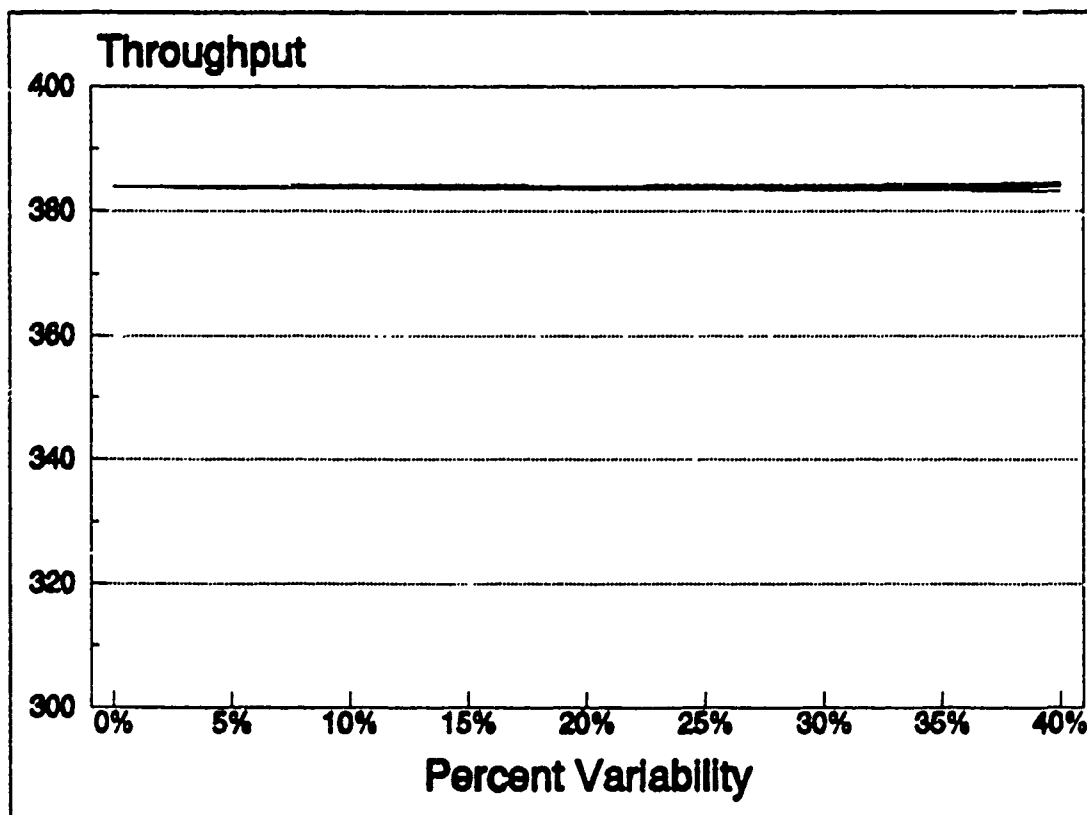
Figure 4-18.   System 3 - JIT - Product 2 Throughput - Process Time Variability

TOC - Drum-Buffer-Rope. The results of this model execution show the characteristic drop in throughput for the lowest WIP level that has been experience with the previous models. The drop for the total throughput of the system goes 864-units down to 813-units (Figure 4-19). As in the previous JIT model this drop is caused by the drop in the Product 1 throughput and not Product 2. If further information is required on this model the program which generated the data is in Appendix F.
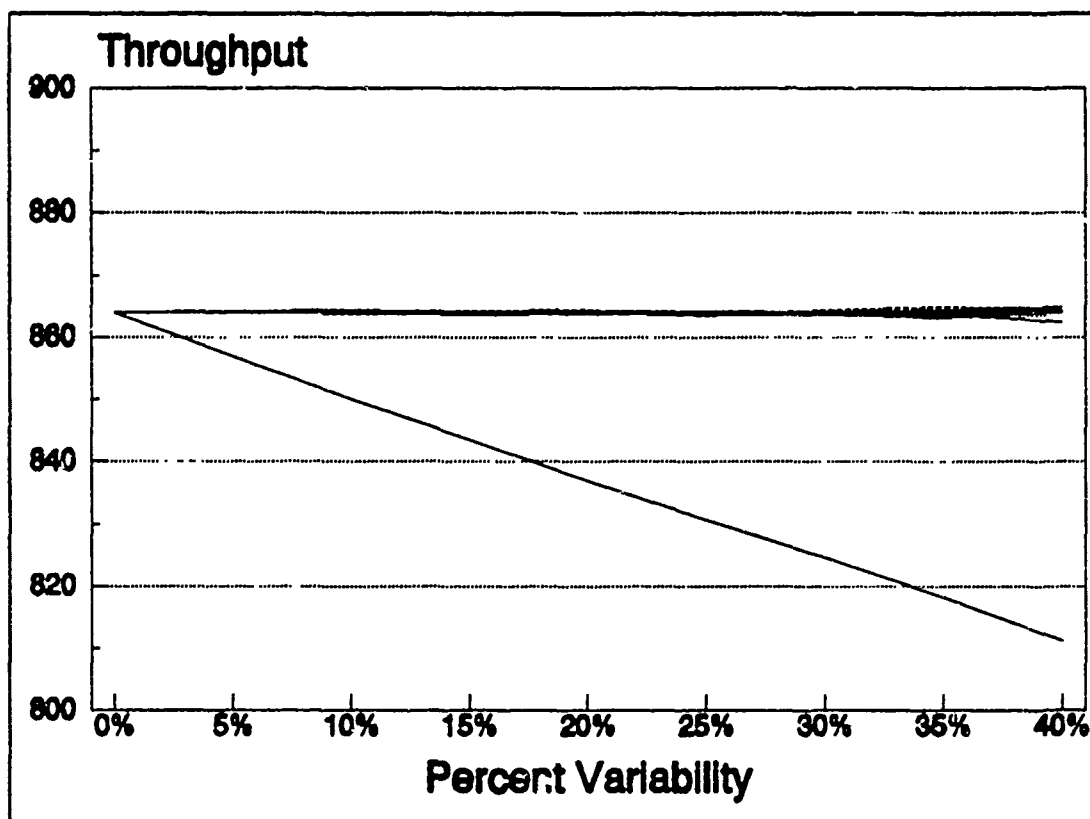
Figure 4-19. System 3 - TOC - Total Throughput - Process Time Variability

The drop in Product 1 throughput for the lowest WIP
level is equal to amount of the drop in the total throughput
(Figure 4-20). This drop is consistent with that
experienced by the previous TOC models for Systems 1 and 2.
An in-depth analysis is made later in this chapter when the
performance of the TOC models of each system is compared.
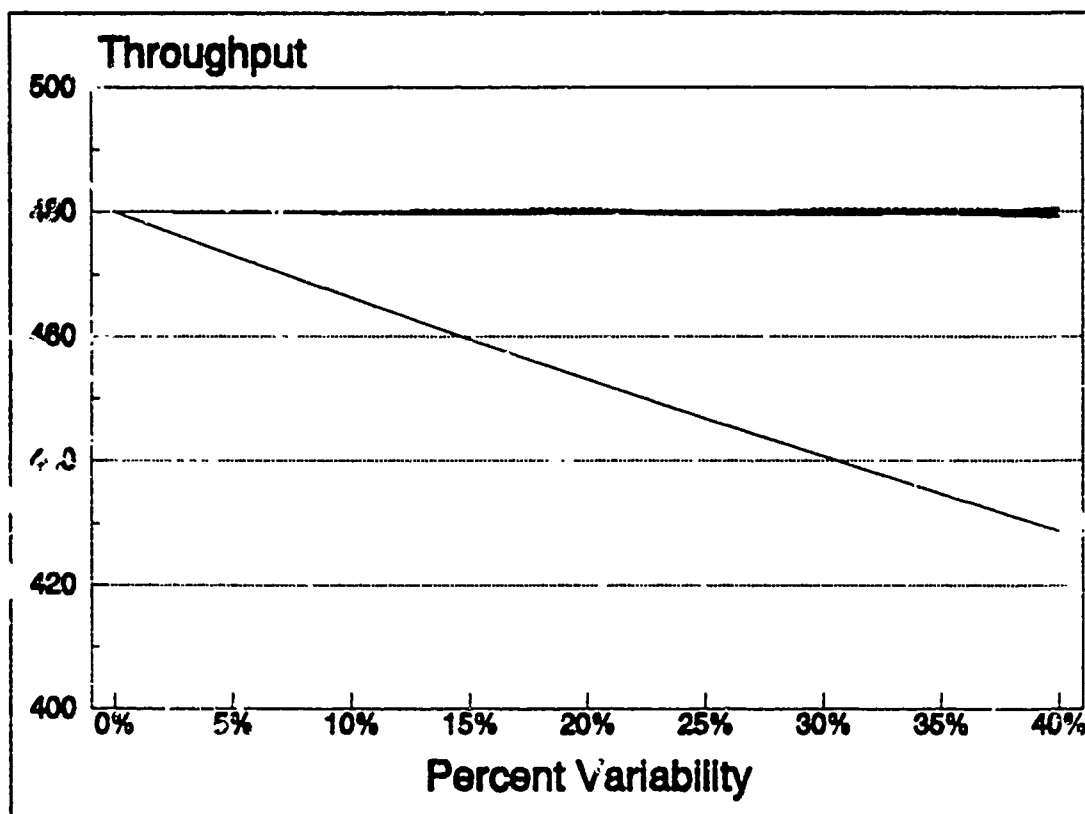
Figure 4-20.   System 3 - TOC - Product 1 Throughput -
                Process Time Variability

The throughput of Product 2 maintained the baseline
value for all levels of WIP and amount of variability
(Figure 4-21). This result is very similar to that obtained

from the previous JIT model, but the cause of this anomaly
can not be the same as the JIT model since the subcomponents
for the TOC model are product designated at the initial
release of material. The lower initial processing times for
both subcomponents may account for the product maintaining
the baseline throughput level.
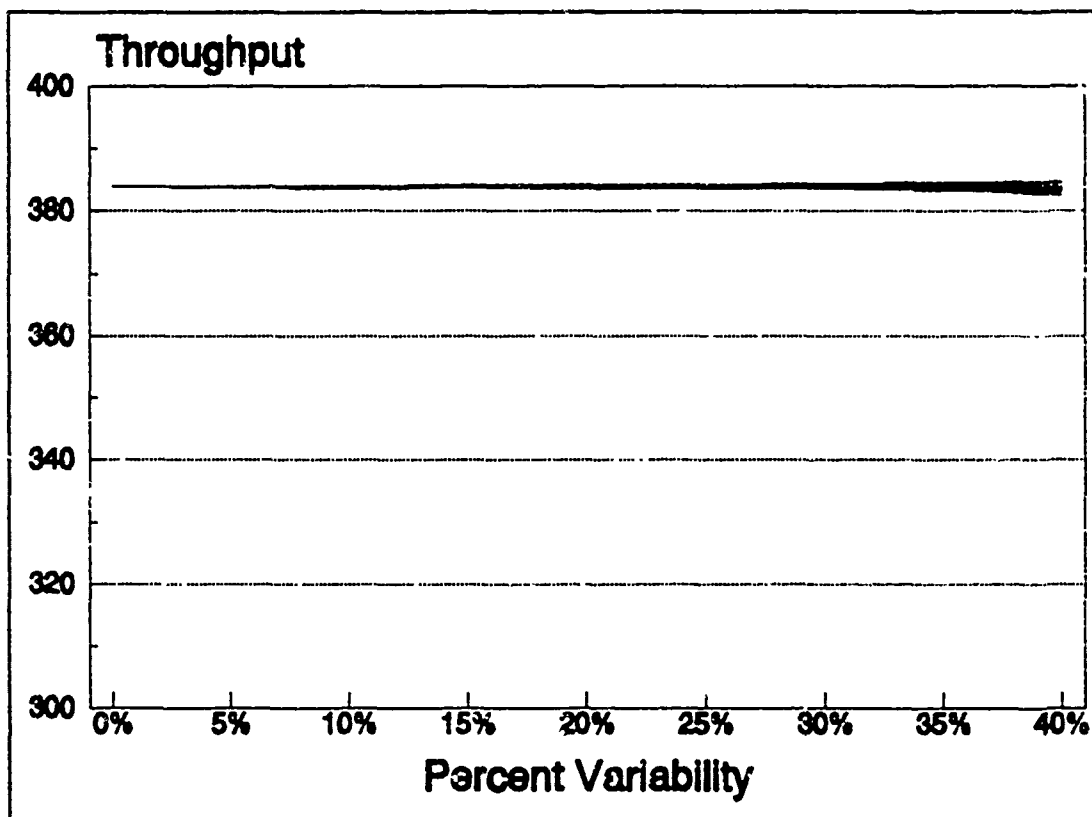
Figure 4-21.   System 3 - TOC - Product 2 Throughput -
Process Time Variability

Comparison of JIT and TOC. The comparison of
these systems is made for only the total throughput of the
system. The baseline throughput for the two products under

each of the scheduling methodologies is the same and the
reaction of the products is independent of the scheduling
methodology selected. Therefore, the comparison of the
total throughput will yield an overall trend in the
performance of the methodologies.

The comparison of the results of these models at low
levels of WIP for this system was complicated by the
buffering systems of the methodologies. The first WIP level
that is common to both models is at the 10-unit level which
for the JIT model is a buffer size of 2, while for the TOC
model it is a buffer size of 3. Previous comparisons at
this level of WIP have yielded inconclusive results since
both models maintain the baseline throughput except for the
lowest level of WIP.

The initial comparison for this system is made the
lowest levels for both systems. The JIT level of WIP is 7-
units, while the throughput for the 6- and 8-unit levels are
shown for the TOC system. Figure 4-22 shows the result of
this comparison including the confidence intervals for each
of the throughput curves. The result is inconclusive since
a direct comparison can not be made with differing WIP
levels. The throughput of the TOC and JIT models both
decrease with increasing variability at the lowest WIP level
for each model while for the next higher level the
throughput is maintained at the baseline level.
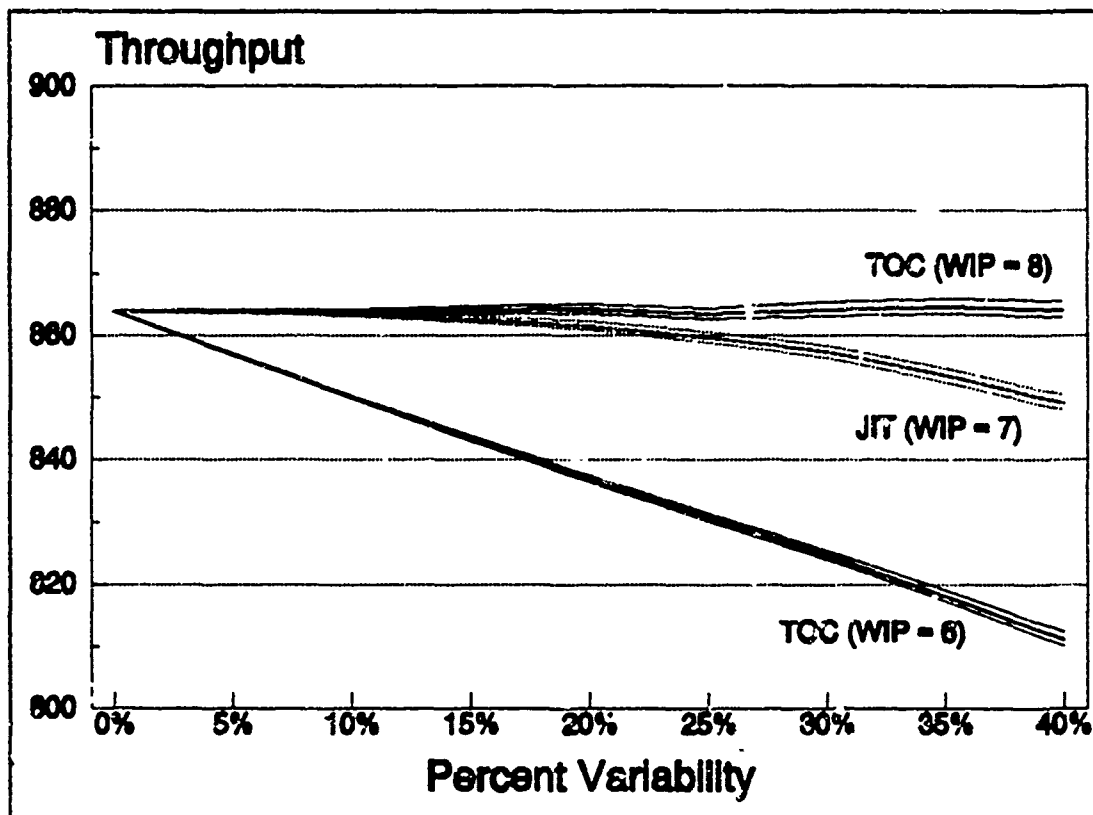
Figure 4-22.   System 3 - Comparison at Lowest WIP Level -
Process Time Variability

The throughput at the 10-unit level of WIP is shown in
Figure 4-23 for both models. As discussed previously both
models maintain the baseline throughput and there is no
statistical difference at the 95 percent confidence level (α
= 0.05). This is the expected result as has been seen in
the comparison of Systems 1 and 2 at levels of WIP above the
very lowest levels for the particular system.
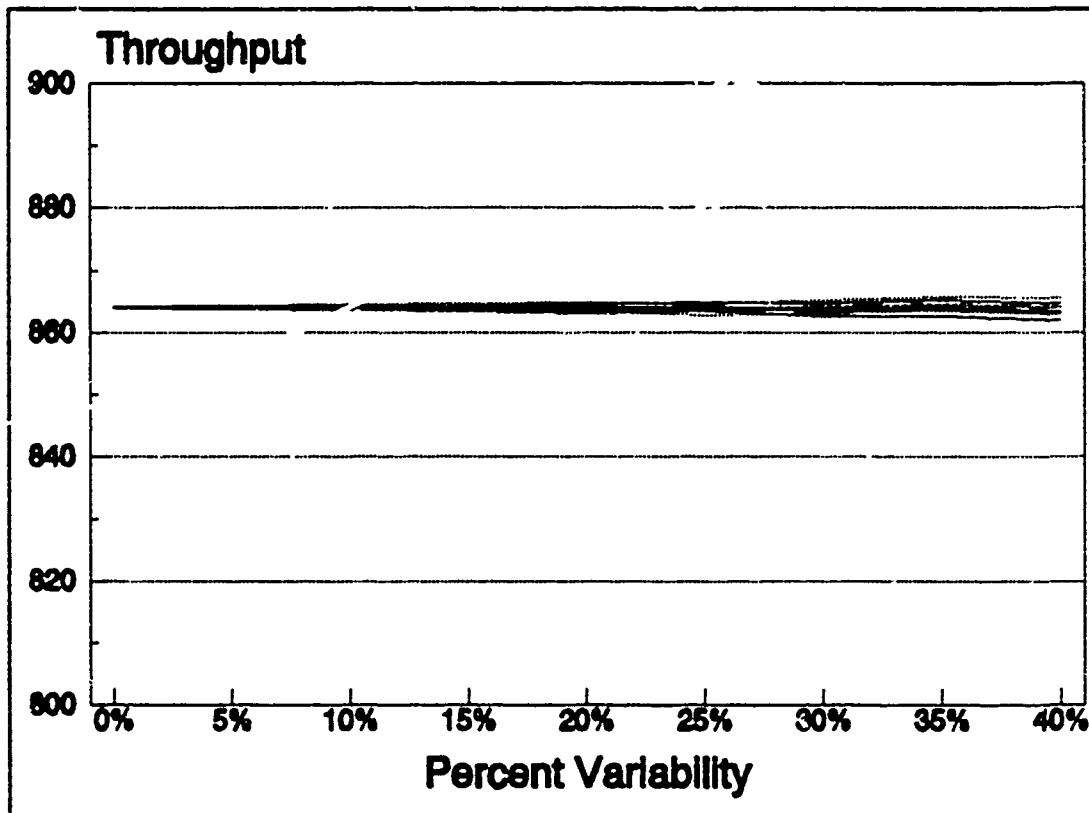
Figure 4-23.   System 3 - Comparison at WIP = 10-units -
Process Time Variability

Probability of Station Failure.   The results of
this portion of the analysis for System 3 are similar to
those obtained from the previous systems when subjected to
increasing probability of process station failure.   The
throughput of both the JIT and TOC models decreases with
increasing probability of failure and the increasing WIP
level does not fully protect the throughput.

JIT with Kanban. The equi-inventory curves
for the total throughput of this model execution show a
greater reaction to the increasing failure rate than those
of previous systems (Figure 4-24). This drop can be
attributed to the increased drop in Product 1 throughput
which is not offset by the decreased drop in Product 2
throughput. If further information is required the program
which generated the data for this analysis is in Appendix E.
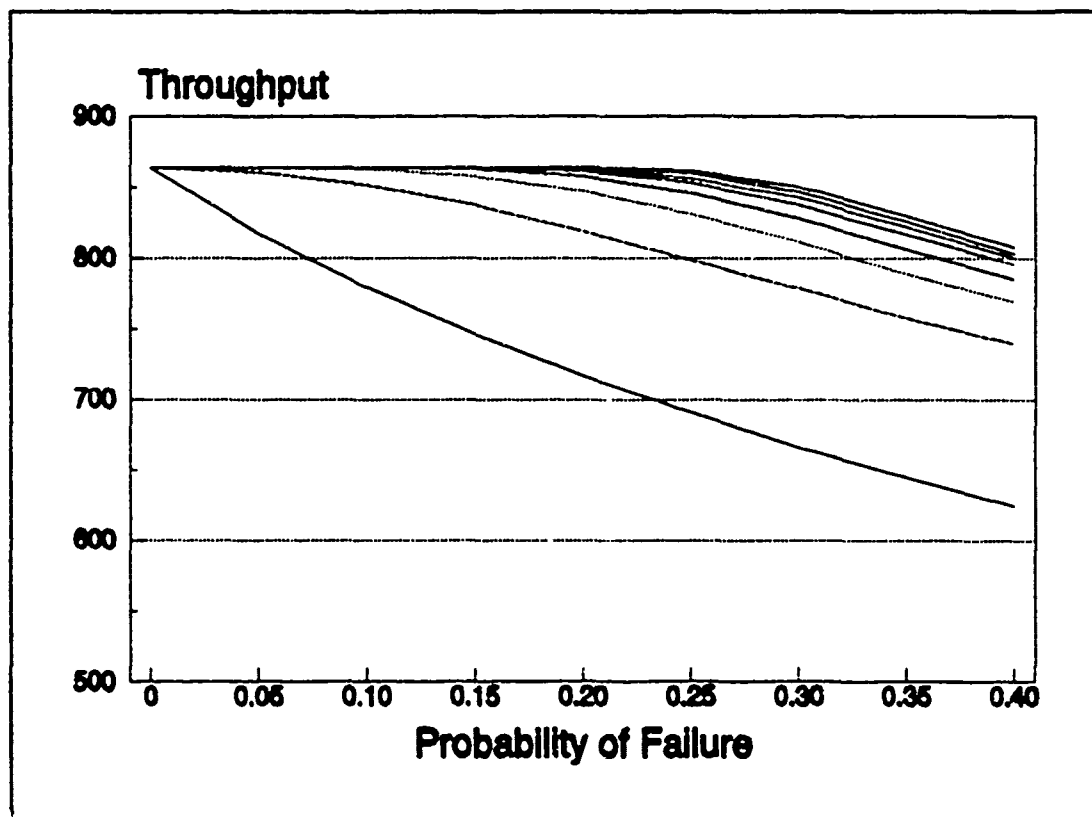
Figure 4-24.   System 3 - JIT - Total Throughput -
               Station Failure

The Product 1 throughput shows an increased drop for all levels of WIP as the probability of failure increases (Figure 4-25). This drop in throughput is greater than that shown by previous JIT system models. The cause of this increased throughput degradation can probably be attributed to the interaction with Product 2 that was exhibited when the model was executed under conditions of process time variability.
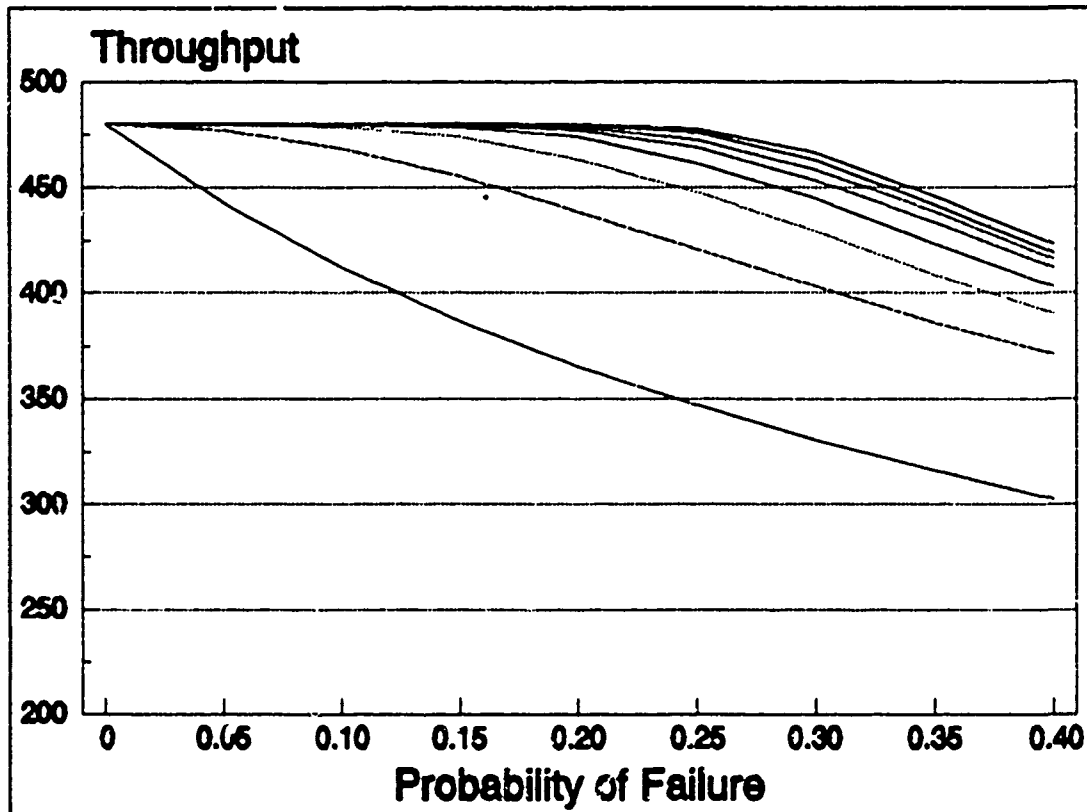
Figure 4-25. System 3 - JIT - Product 1 Throughput - Station Failure

As stated previously, the reaction of Product 2 to increasing probability of failure was not as severe as seen in the model executions of previous systems. Product 2 used its interaction with Product 1 to maintain the baseline throughput for all probability levels except at the lower levels of WIP. Figure 4-26 shows the reaction of Product 2 to increasing probability of failure for various levels of WIP.
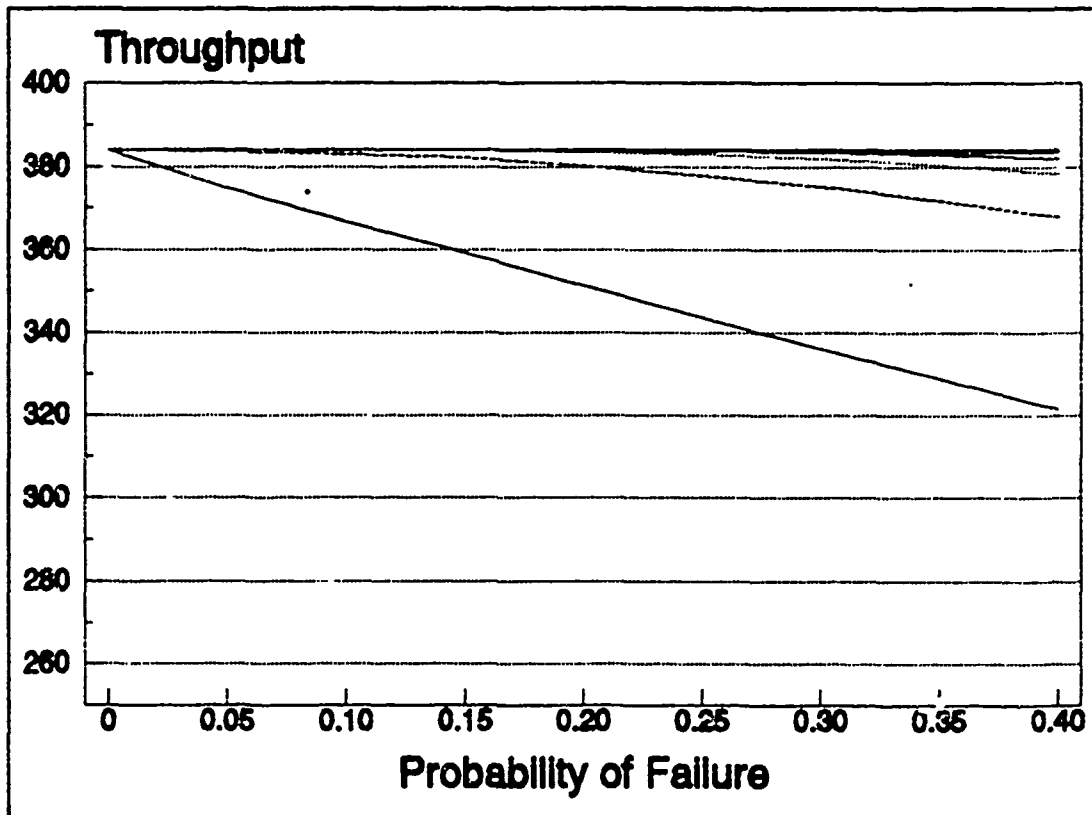
Figure 4-26. System 3 - JIT - Product 2 Throughput - Station Failure

TOC - Drum-Buffer-Rope. The reaction of the
TOC model to increasing probability of failure is very
similar to that seen in previous TOC model executions. The
total throughput of the system is degraded for all levels of
WIP as the probability of failure increases (Figure 4-27).
The increase in WIP can not fully protect the throughput of
the system although an increase in throughput is seen as the
WIP increases. The convergence of the equi-inventory curves
into a single curve is also the same as that seen with
previous systems.
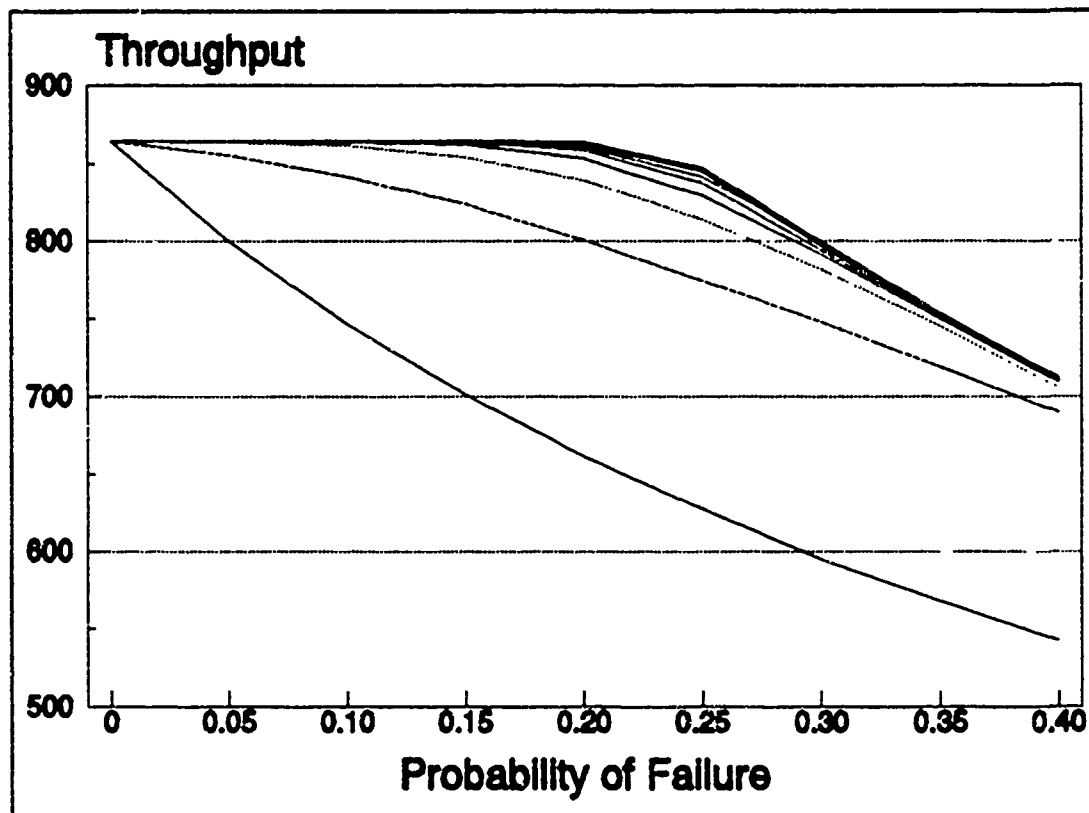
Figure 4-27. System 3 - TOC - Total Throughput -
Station Failure

If further information is required the program which generated the data used in this analysis is in Appendix F.

The throughput of Product 1 is similar to the general shape of the curves seen in previous TOC model executions for increasing probability of station failure. The curves again converge on a single maximum curve as WIP increases (Figure 4-28). The equi-inventory curves exhibit the same spread between each level of WIP as was seen in previous model executions.
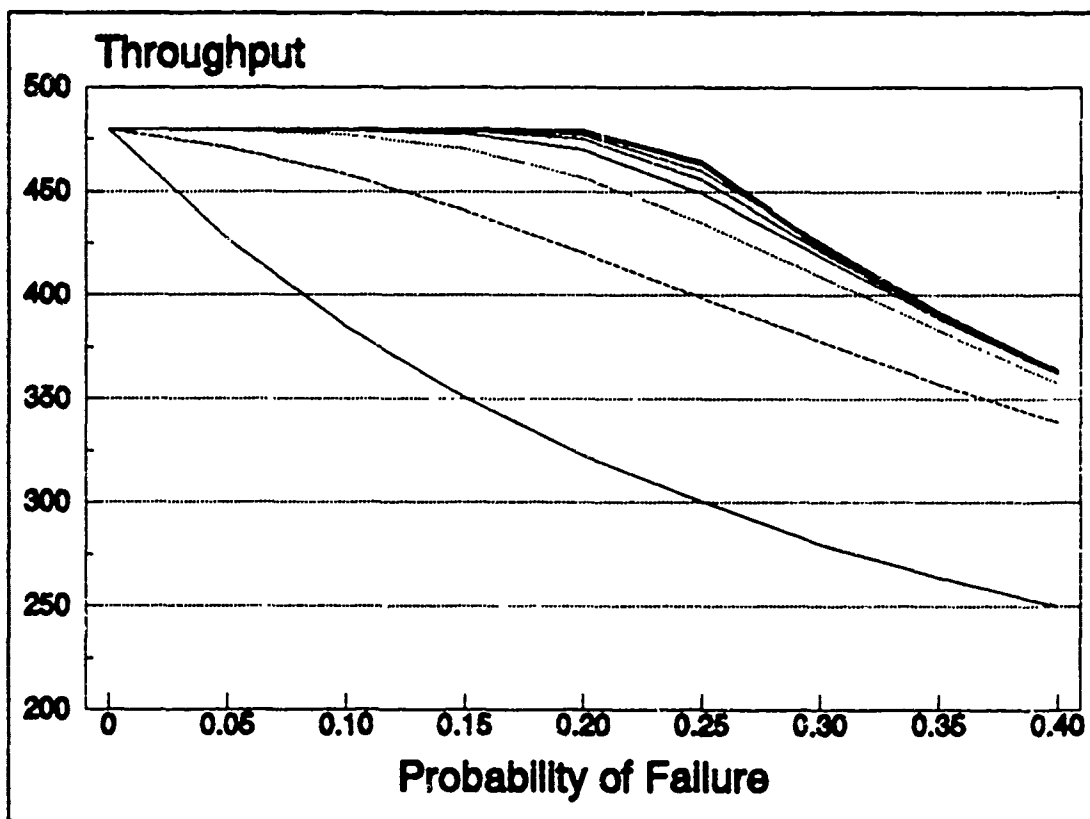
Figure 4-28.   System 3 - TOC - Product 1 Throughput - Station Failure

The throughput of Product 2 differs from that which has been characteristic of other model executions. While the throughput is degraded by increasing probability of failure the equi-inventory curves converge on a single curve much faster than previous systems (Figure 4-29). This convergence is probably a characteristic of the relationship of the processing times of the stations in the flow of Product 2 since the subcomponents of the two products are independent in the TOC model (Appendix F).
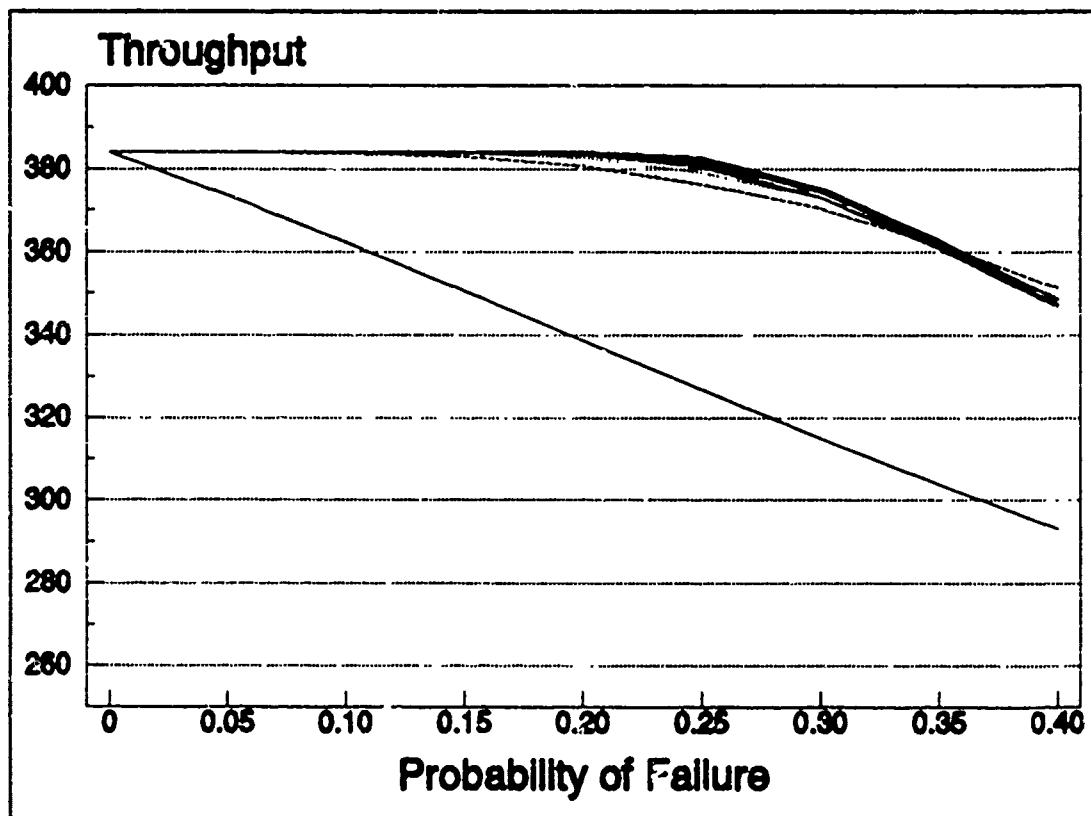
Figure 4-29.   System 3 - TOC - Product 2 Throughput - Station Failure

Comparison of JIT and TOC. The comparison of
these scheduling methodologies will, as in the process time
variability comparison, be made only for the total
throughput of the system. The comparison of the throughput
of the individual products is unnecessary since this
research study is concerned only with the general reaction
of the scheduling methodology to specific system
variabilities.

The comparison at the lower levels of WIP is again
complicated by the buffering of the models. The first level
of WIP which is common to both systems is at the 10-unit
level which is significantly above the level at which
comparisons were made for previous systems. As was done in
the comparison for process time variability the graph for
the lowest WIP level is presented in Figure 4-30. While the
analysis of this graph is inconclusive as to which
methodology yields the greater throughput for the lowest
level of WIP, a general knowledge can be gained from a
visual comparison at these low levels. The TOC model at the
8-units of WIP level yields a greater throughput and
provides nearly the baseline throughput for a greater range
of failure probability values. The throughput of the TOC
model at a WIP level of 8-units also provides an increase in
throughput of approximately 50 units over the JIT model for
only an increase of 1-unit of WIP.
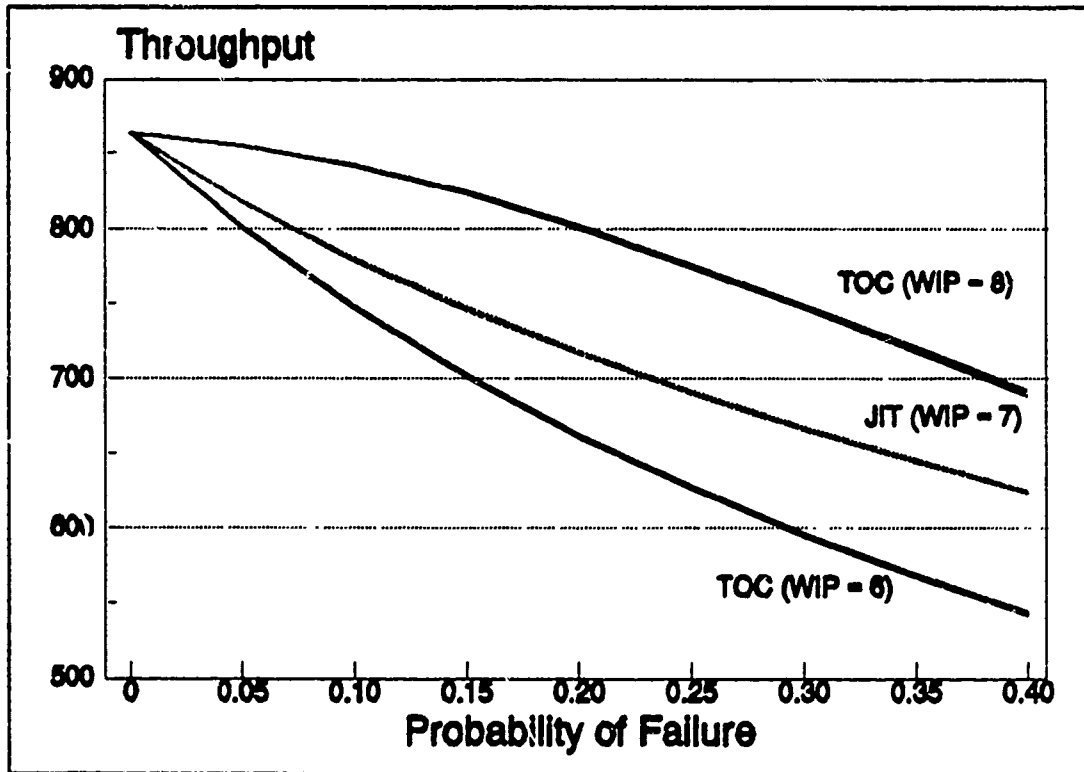
Figure 4-30.   System 3 - Comparison at Lowest WIP Level -
Station Failure

The comparison at the 10-unit level of WIP is also
inconclusive for this particular model as the throughput
curves cross at approximately the 0.35 level of probability
(Figure 4-31). This indicates that for level of failure
less than 0.35 the TOC model provides increased throughput
while at higher failure rates the JIT model outperfor.ᵤ the
TOC model. This type of result was not expected from the
analysis and has not been experienced in previous systems.
The interaction of the two products produced by the system
could account for this result, due to the differences in the
internal buffering of the methodologies.
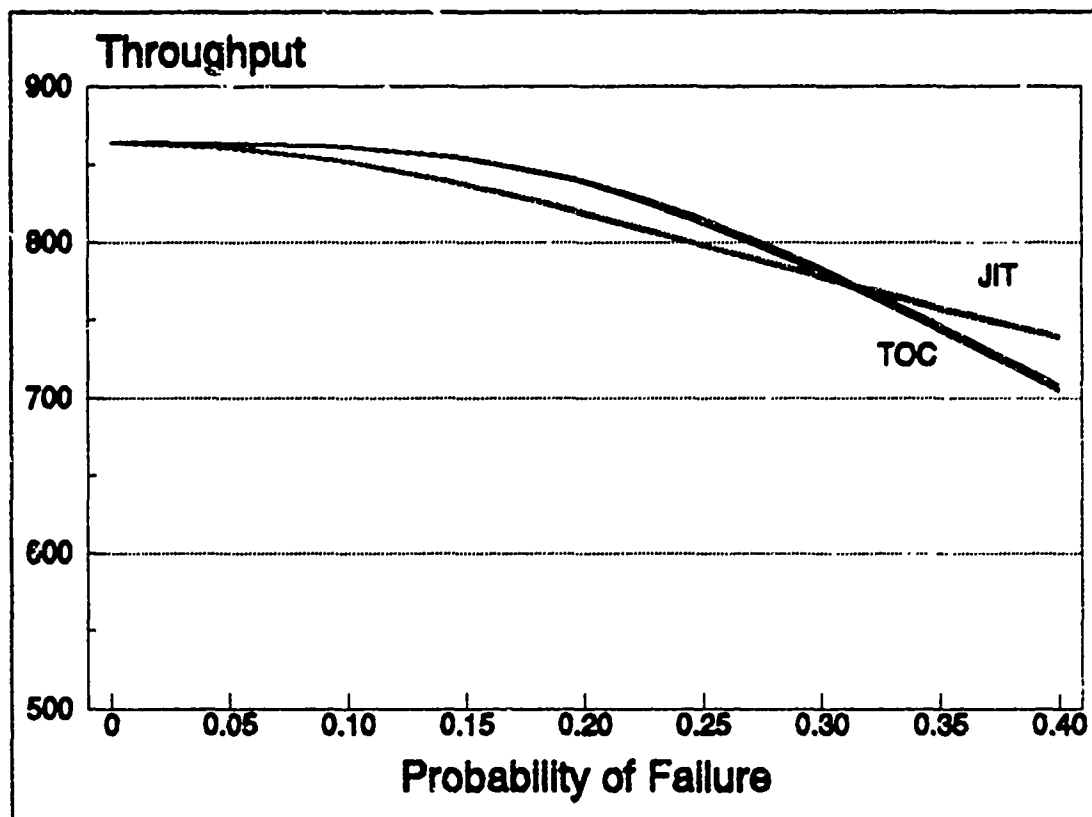
Figure 4-31.   System 3 - Comparison at WIP = 10-units -
               Station Failure

Comparison of TOC Results. The comparison of the TOC

models for each of the three systems are presented to

determine if there is any difference due to the structure of

the system on the throughput achieved by the TOC scheduling

methodology. This analysis will be conducted separately for

both types of system variability to determine if the type of

variability affects the throughput. The total throughput of

Systems 1 and 2 will be used as the critical path through

both of these systems is identical. The Product 1

throughput of System 3 will be utilized in the comparison

since its critical path is again the same as that of the total throughput of Systems 1 and 2.

Process Time Variability. The initial comparison that is conducted is at a buffer size of one unit for each of the systems. This buffer size yields a WIP level of three units for Systems 1 and 2, and six units total for System 3. These 6 units for System 3 are divided evenly between Products 1 and 2. It can be seen in Figure 4-32 that at this level of WIP there is no difference between the systems due to differing flow patterns. This is the expected result for the analysis since there is no difference between the systems in the amount of time required to produce a product.
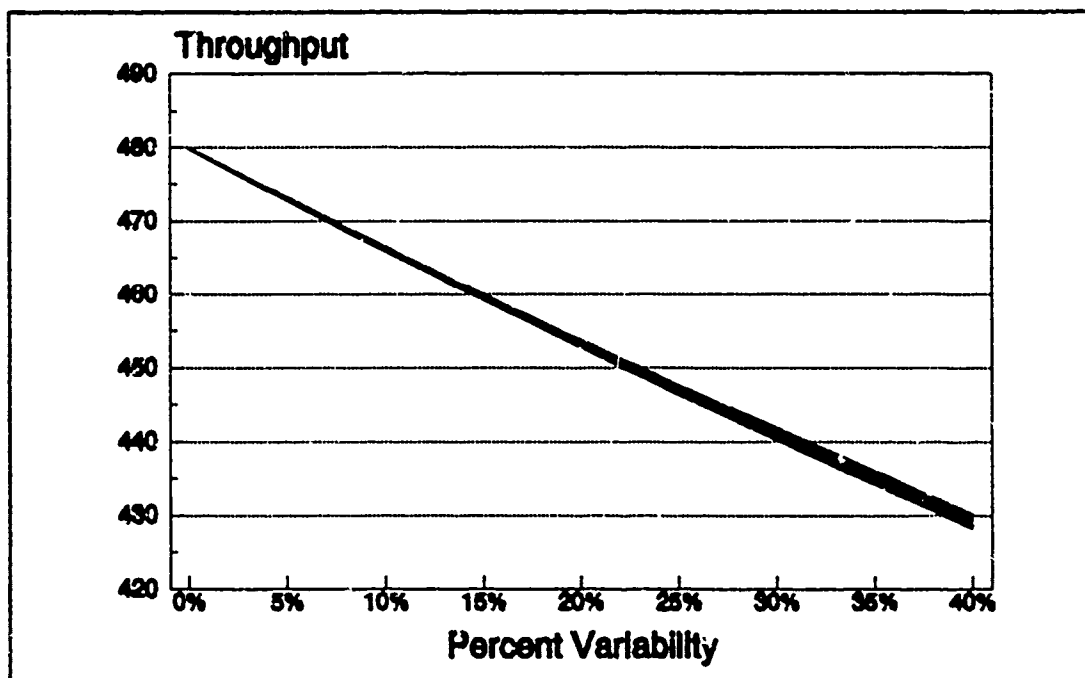
Figure 4-32. System Comparison - Buffer Size = 1-unit - Process Time Variability

The next comparison for process time variability will be conducted at a buffer size of two units. It can be seen in Figure 4-33 that all of the systems produce the baseline throughput for all levels of process time variability and there is no statistical difference between the systems at the 95 percent confidence level ($\alpha = 0.05$). This result is again what was expected and as could be determined from the previous presentations of the results of the model executions.
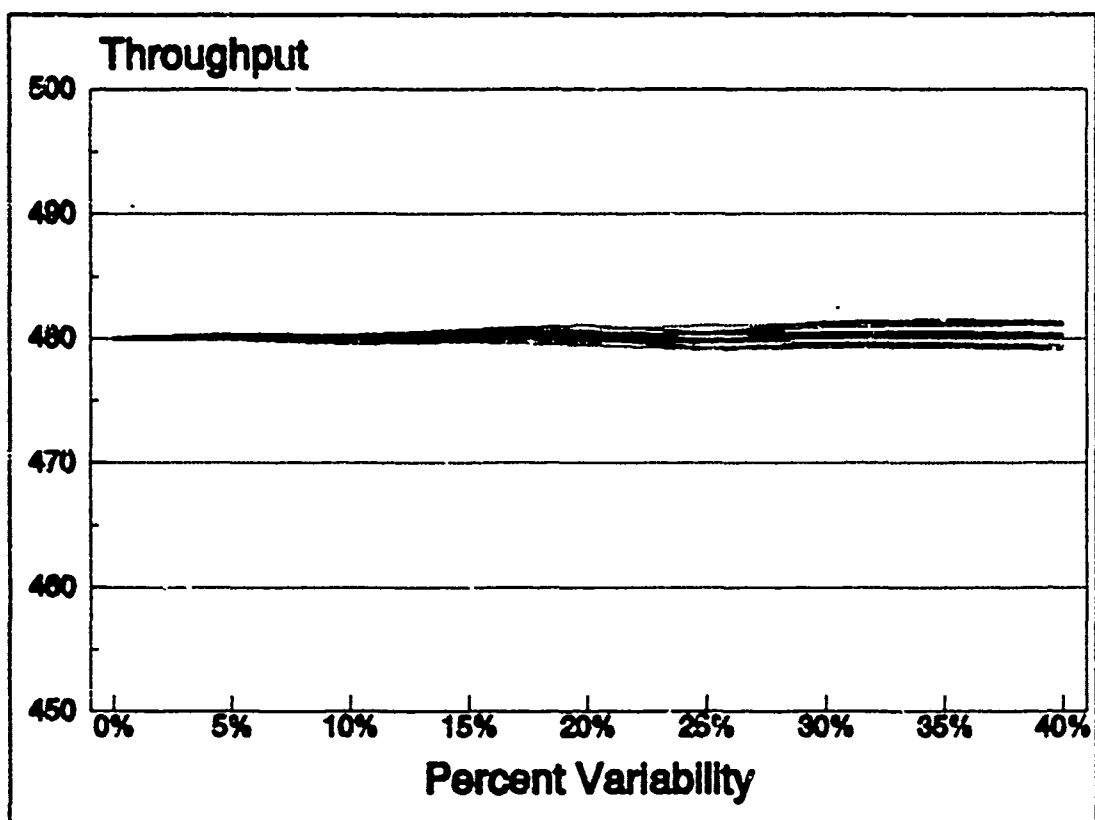
Figure 4-33. System Comparison - Buffer Size = 2-units - Process Time Variability

Probability of Station Failure. This comparison
is conducted at the same buffer sizes as the previous
comparison for process time variability. The results of
this comparison are more interesting than that of the
process time variability due to the interaction of
assembling the subcomponents of the product produced by
Systems 2 and 3. At the buffer size of 1-unit, Figure 4-34
shows that the throughput of System 1 is statistically
greater for all failure probabilities than the throughput of
Systems 2 and 3. There is no statistical difference (α =
0.05) between the throughput curves for Systems 2 and 3.
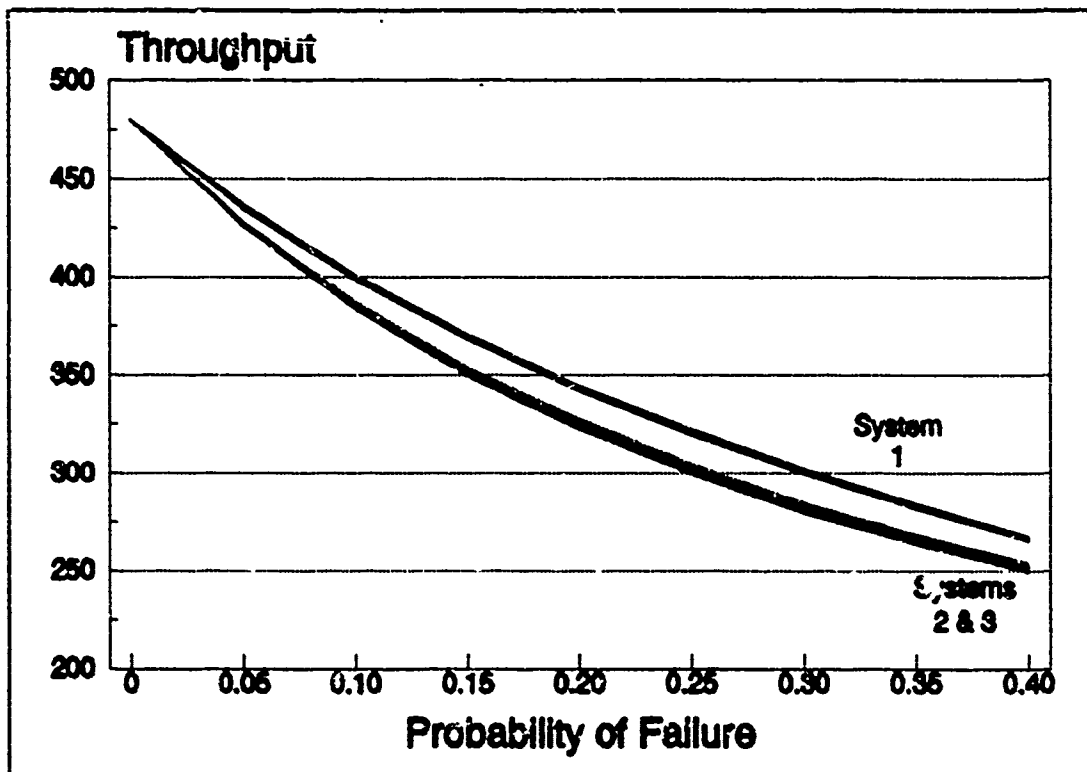
Figure 4-34. System Comparison - Buffer Size = 1-unit -
Station Failure

Figure 4-35 shows the comparison of the systems at a
buffer size of two units. In this case Systems 1 and 2
provide the same throughput with no statistical difference
while System 3 throughput is statistically lower for all
probability of failure values. This situation could be
caused by the interaction of Product 1 with Product 2 in
obtaining access to the common processing stations.
However, the difference between the throughput curves, while
statistically significant at the 95 percent confidence level
($\alpha = 0.05$), is still fairly small when compared to the
overall throughput of the models.
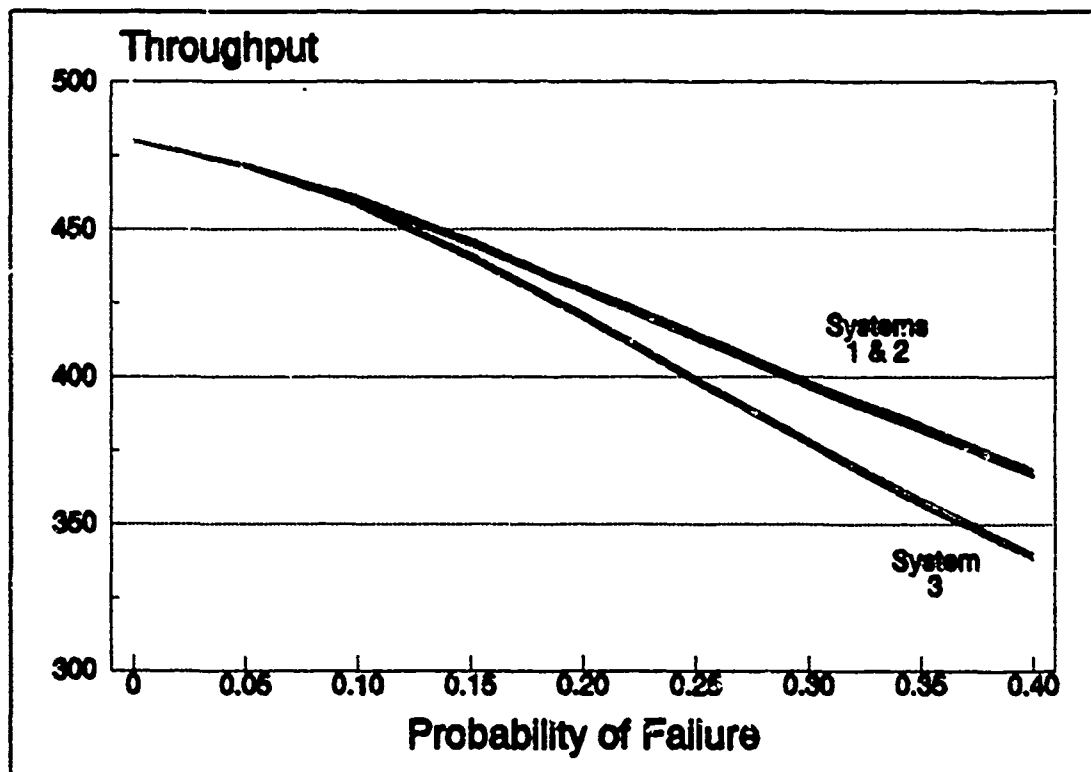
Figure 4-35. System Comparison - Buffer Size = 2-units -
Station Failure

V. Conclusions and Recommendations

Introduction

The original intent of this research effort was to answer two specific research questions. The research questions which were posed in Chapter 1 are:

1. Does TOC's Drum-Buffer-Rope (DBR) technique resolve the throughput problems associated with system variability in JIT scheduled systems?

2. Is the improvement in throughput achieved in TOC systems dependent on the flow path of the production system under consideration?

The analysis conducted in the first portion of Chapter 4 where Just-in-Time (JIT) and TOC models were compared addressed the first research question. The comparison conducted between the results of the TOC models at the end of Chapter 4 addresses the second question.

Conclusions

Research Question 1. The results of the analysis conducted in the previous chapter lead to the conclusion that for Systems 1 and 2 the Theory of Constraints (TOC) provides superior stability of throughput under process time variability. When these systems were subjected to process station failure the TOC models provided superior throughput for equivalent amounts of work-in-process (WIP). The results of the System 3 models are inconclusive, but the TOC model does appear to provide equivalent throughput to the JIT model independent of the type of variability.

The analysis conducted under conditions of process time variability show that for System 1 (Figures 4-3 and 4-4) and System 2 (Figure 4-12) the TOC scheduled model provided better throughput stability than the JIT system. As the process time variability percentage increased the JIT scheduled models throughput began to drop below the baseline throughput value. The expanded analysis (Figure 4-4) done on System 1 shows that the decrease continues as the variability increased above the normal level of this study. The results of System 3 showed no differences between JIT and TOC systems. Figure 4-23 shows that at the 10-unit level of WIP both scheduling methodologies provided the baseline throughput over the entire range of variability.

The analysis conducted for process station failure is more conclusive than the process time variability analysis. For Systems 1 and 2 (Figures 4-8, 4-9, and 4-15) the TOC scheduled model outperformed the JIT model by approximately 25-units over a large range of failure probabilities. The TOC system did experience degradation of throughput due to process station failure, but it was not as significantly affected by the failure as the JIT system. The results for System 3 were more inconclusive for process station failure than for process time variability. At the 10-unit level of WIP the TOC system outperformed the JIT system marginally up to the 0.30 probability level and then the JIT system outperformed the TOC system for high probability levels (Figure 4-31). This anomaly could have been due to product

interaction differences caused by the buffering of the scheduling methodologies. In the JIT system the common subcomponent was released into the system as demanded by the first buffer in its flow path. The TOC system released the common subcomponent as required by the particular product buffer. The common subcomponent in the TOC system was designated at release as to the product requiring the subcomponent. This was not done in the JIT system due to the nature of the JIT buffering.

Research Question 2. The conclusion that can be derived from the results of the process time variability comparison is that these results are not affected by the flow path of the system. The results of the process station failure comparison lead to the conclusion that there is some dependence on the flow path of the system.

Figures 4-32 and 4-33 both show that there is no statistical difference between the throughputs achieved by any of the different systems. The throughput achieved for each of the variability percentages for buffer sizes of one and two units are statistically identical.

When the systems were compared under conditions of process station failure the results were mixed. For a buffer size of one unit System 1 outperformed the other two systems which had statistically identical throughput curves. When the buffer size was increased to two units Systems 1 and 2 had statistically identical throughput curves while System 3 provided significantly lower throughput.

Recommendations

There is significant research that needs to be conducted before the full characteristics of TOC can be verified. This is one of the first independent research efforts that has been conducted on TOC so the opportunity for future research is unlimited.

One of the anomalies that was documented by this research effort was the convergence of the equi-inventory curves on a single curve when subjected to process station failure (Figures 4-6, 4-7, 4-14, and 4-27). This convergence should be investigated further to determine the exact cause of the phenomenon.

The analysis of System 3 in this research also illustrated several product interaction effects that could have a significant impact on an actual production system. Few production systems produce a single product so the determination of the causes and effects of product interaction would significantly aid the production manager in effectively scheduling product flow.

The final area in which this research should be expanded is into an actual production system. The researcher could develop and validate a simulation model using the current production scheduling system. This model could then be used to compare the effects of JIT and TOC scheduling on an actual production system without the cost

and disruption caused by experimenting with the actual system.

This research effort is simply the starting point for many future research efforts that are needed by before any conclusive statement can be made as to the relative effectiveness of the Theory of Constraints scheduling methodology. TOC appears to have partially resolved some of the variability problems associated with JIT, but a great deal of future research is required using both theoretical and actual production systems.

```
        REALLOCATE COM,150000
*
        SIMULATE
*
*       Variable Declaration Section
*
**********************************************
*
*       Variable Definitions
*
**********************************************
*
*       &I, &J  - Loop Counters
*       &BFSZ   - Defined Buffer Size
*       &BUFF1  - Buffer Counter for Buffer 1
*       &BUFF2  - Buffer Counter for Buffer 2
*       &BUFF3  - Buffer Counter for Buffer 3
*       &POF    - Defined Probability of Station Failure
*       &DELT   - Percent Variability of Processing Time
*       &MN1    - Mean Processing Time for Station 1
*       &MN2    - Mean Processing Time for Station 2
*       &MN3    - Mean Processing Time for Station 3
*       &MN4    - Mean Processing Time for Station 4
*       &MEAN1  - Calculated Actual Processing Time for Station 1
*       &MEAN2  - Calculated Actual Processing Time for Station 2
*       &MEAN3  - Calculated Actual Processing Time for Station 3
*       &MEAN4  - Calculated Actual Processing Time for Station 4
*
**********************************************
*
        INTEGER    &I,&J
        INTEGER    &BFSZ,&BUFF1,&BUFF2,&BUFF3
        REAL       &POF,&MN1,&MN2,&MN3,&MN4
        REAL       &MEAN1,&MEAN2,&MEAN3,&MEAN4,&DELT
*
**********************************************
*
*       Declaration of Output File
*
**********************************************
*
 OUTA   FILEDEF    'THSJ1A1.OUT'
*
```

```
*
*
*       Variable Assignment Section
*
*
**********************************************
*
*       Definition of Mean Processing Times
*
**********************************************
*
        LET       &MN1=7.5
        LET       &MN2=12.5
        LET       &MN3=20.0
        LET       &MN4=5.0
*
*       GPSS/H Block Section
*
        GENERATE  0,,0           Generate a XACT whenever the
*                                input buffer is empty
        SEIZE     INBUF          Enter the input buffer
*
*       Beginning of product flow
*
        TEST NE   &BUFF1,&BFSZ   Test Buffer 1 for capacity
        BLET      &BUFF1=&BUFF1+1 Increment Buffer 1 counter
        TEST E    F(STA1),0      See if Sta 1 is busy
        RELEASE   INBUF          Release the input buffer
*
*
        QUEUE     SYSQ           Enter the system queue
*
*********************************************
*
*       Process 1
*
*********************************************
*
        SEIZE     STA1                 Seize Station 1
        TEST LE   FRN(2),&POF,STRT1    Determine if Sta 1 fails
        FUNAVAIL  PRCS1                Make Sta 1 unavailable
        ADVANCE   2*&MN1               Station unavailable delay time
        FAVAIL    PRCS1                Make Sta 1 available
*
STRT1  SEIZE     PRCS1                Seize Processing Sta 1
*
        BLET      &MEAN1=&MN1+(((FRN(3,-0.5)*2)*&MN1*&DELT)
        ADVANCE   &MEAN1
        RELEASE   PRCS1
        RELEASE   STA1                 Release Station
```

A-2

```
*
*************************
*
*      Buffer 1
*
*************************
*
        ENTER     BFFR1                  Enter Buffer 1
        TEST NE   &BUFF2,&BFSZ           Check capacity of Buffer 2
        BLET      &BUFF2=&BUFF2+1        Increment Buffer 2 counter
        TEST E    F(STA2),0              Check status of Sta 2
        BLET      &BUFF1=&BUFF1-1        Decrement Buffer 1 counter
        LEAVE     BFFR1                  Leave Buffer 1
*
*
***********************************
*
*      Process 2
*
***********************************
*
        SEIZE     STA2
        TEST LE   FRN(4),&POF,STRT2
        FUNAVAIL  PRCS2
        ADVANCE   2*&MN2
        FAVAIL    PRCS2
*
 STRT2 SEIZE      PRCS2
*
        BLET      &MEAN2=&MN2+(((FRN(5)-0.5)*2)*&MN2*&DELT)
        ADVANCE   &MEAN2
        RELEASE   PRCS2
        RELEASE   STA2
*
*************************
*
*      Buffer 2
*
*************************
*
        ENTER     BFFR2
        TEST NE   &BUFF3,&BFSZ
        BLET      &BUFF3=&BUFF3+1
        TEST E    F(STA3),0
        BLET      &BUFF2=&BUFF2-1
        LEAVE     BFFR2
*
```

```
*
*********************************
*
*     Process 3
*
*********************************
*
      SEIZE     STA3
*
      SEIZE     PRCS3
*
      BLET      &MEAN3=&MN3+(((FRN(7)-0.5)*2)*&MN3*&DELT)
      ADVANCE   &MEAN3
      RELEASE   PRCS3
      RELEASE   STA3
*
***************************
*
*     Buffer 3
*
***************************
*
      ENTER     BFFR3
      TEST E    F(STA4),0
      BLET      &BUFF3=&BUFF3-1
      LEAVE     BFFR3
*
*
*********************************
*
*     Process 4
*
*********************************
*
      SEIZE     STA4
      TEST LE   FRN(8),&POF,STRT4
      FUNAVAIL  PRCS4
      ADVANCE   2*&MN4
      FAVAIL    PRCS4
*
STRT4 SEIZE     PRCS4
*
      BLET      &MEAN4=&MN4+(((FRN(9)-0.5)*2)*&MN4*&DELT)
      ADVANCE   &MEAN4
      RELEASE   PRCS4
      RELEASE   STA4
*
      DEPART    SYSQ
*
PDCT  TERMINATE 0          Terminate the XACT
```

```
*
*        Model Timing
*
*     Run model for 40 hour week
*
         GENERATE    60.*8.*5.
         TERMINATE   1
*
*****************************************
*
*        GPSS/H Control Statements
*
*****************************************
*
*        Process Time Variability (&DELT) initialized.
*        The values in successive models ranged between
*        0.0 and 0.40 in increments of 0.05.
*
         LET         &DELT=0.0
*
*        Probability of Station Failure (&POF) initialiized.
*        The values in successive models ranged between
*        0.0 and 0.40 in increments of 0.05.
*
         LET         &POF=0.0
*
         LET         &BFSZ=1          Buffer size initialized
         DO          &J=1,8
  BFFR1  STORAGE     &BFSZ
  BFFR2  STORAGE     &BFSZ            Buffer storages set to &BFSZ
  BFFR3  STORAGE     &BFSZ
         DO          &I=1,100         100 iterations at each set-point
         LET         &BUFF1=0
         LET         &BUFF2=0
         LET         &BUFF3=0
         START       1,NP             40 hour warm-up period
         RESET
         START       4,NP             160 hour data gathering period
*
*        Output of data to file
*
         PUTPIC      FILE=OUTA,(&DELT,&POF,N(PDCT),QM(SYSQ))
  ***.**  *.**   ****   ***
         CLEAR
         ENDDO
         LET         &BFSZ=&BFSZ+1  Increment Buffer Size
         ENDDO
         END
```

Appendix B: System 1 - TOC - GPSSH Program Listing

```
        REALLOCATE COM,150000
*
        SIMULATE
*
*
*       Variable Declaration Section
*
*****************************************
*
*       Variable Definitions
*
*****************************************
*
*       &I, &J  - Loop Counters
*       &BFSZ   - Defined Buffer Size
*       &BUFF   - Buffer Counter for System Buffer
*       &POF    - Defined Probability of Station Failure
*       &DELT   - Percent Variability of Processing Time
*       &MN1    - Mean Processing Time for Station 1
*       &MN2    - Mean Processing Time for Station 2
*       &MN3    - Mean Processing Time for Station 3
*       &MN4    - Mean Processing Time for Station 4
*       &MEAN1  - Calculated Actual Processing Time for Station 1
*       &MEAN2  - Calculated Actual Processing Time for Station 2
*       &MEAN3  - Calculated Actual Processing Time for Station 3
*       &MEAN4  - Calculated Actual Processing Time for Station 4
*
*****************************************
*
*
        INTEGER    &I,&J
        INTEGER    &BFSZ,&BUFF
        REAL       &POF,&MN1,&MN2,&MN3,&MN4
        REAL       &MEAN1,&MEAN2,&MEAN3,&MEAN4,&DELT
*
*****************************************
*
*       Declaration of Output File
*
*****************************************
*
*
 OUTA   FILEDEF    'TESC1A1.OUT'
*
*
*
```

```
*
*         Variable Assignment Section
*
**************************************************
*
*         Definition of Mean Processing Times
*
**************************************************
*
*
*
          LET       &MN1=7.5
          LET       &MN2=12.5
          LET       &MN3=20.0
          LET       &MN4=5.0
*
*         GPSS/H Block Section
*
          GENERATE  0,,0           Generate a XACT whenever the
*                                  input buffer is empty
          SEIZE     INBUF          Enter the input buffer
*
*
*         Beginning of product flow
*
*
          TEST NE   &BUFF,&BFSZ    Test for room in Buffer
          BLET      &BUFF=&BUFF+1  Increment Buffer Counter
          TEST E    F(STA1),0      Test Status of Station 1
          RELEASE   INBUF          Leave Input buffer
*
*
          QUEUE     SYSQ           Enter System queue
*
**********************************************
*
*         Process 1
*
**********************************************
*
*
          SEIZE     STA1                 Enter Station 1
          TEST LE   FRN(2),&POF,STRT1    Test Failure of Sta 1
          FUNAVAIL  PRCS1                Make Sta 1 unavailable
          ADVANCE   2*&MN1               Repair Station 1
          FAVAIL    PRCS1                Make Sta 1 Available
*
STRT1 SEIZE       PRCS1                  Enter Pressing of Sta 1
*
          BLET      &MEAN1=&MN1+(((FRN(3)-0.5)*2)*&MN1*&DELT)
          ADVANCE   &MEAN1
          RELEASE   PRCS1
          RELEASE   STA1                 Leave Station 1
```

```
********************************

*

*        Process 2

*

********************************

*

*

        SEIZE       STA2
        TEST LE     FRN(4),&POF,STRT2
        FUNAVAIL    PRCS2
        ADVANCE     2*&MN2
        FAVAIL      PRCS2
*
 STRT2  SEIZE       PRCS2
*
        BLET        &MEAN2=&MN2+(((FRN(5)-0.5)*2)*&MN2*&DELT)
        ADVANCE     &MEAN2
        RELEASE     PRCS2
        RELEASE     STA2
*
********************************

*

*        System Buffer

*

********************************

*

        ENTER       BFFR            Enter the system buffer
        TEST E      F(STA3),0       Check status of Sta 3
        BLET        &BUFF=&BUFF-1   Decrement buffer counter
        LEAVE       BFFR            Leave the system buffer
*
*
********************************

*

*        Process 3

*

********************************

*

*

        SEIZE       STA3
*
        SEIZE       PRCS3
*
        BLET        &MEAN3=&MN3+(((FRN(7)-0.5)*2)*&MN3*&DELT)
        ADVANCE     &MEAN3
        RELEASE     PRCS3
        RELEASE     STA3
*
```

```
*********************************
*
*        Process 4
*
*********************************
*
        SEIZE     STA4
        TEST LE   FRN(8),&POF,STRT4
        FUNAVAIL  PRCS4
        ADVANCE   2*&MN4
        FAVAIL    PRCS4
*
 STRT4  SEIZE     PRCS4
*
        BLET      &MEAN4=&MN4+(((FRN(9)-0.5)*2)*&MN4*&DELT)
        ADVANCE   &MEAN4
        RELEASE   PRCS4
        RELEASE   STA4
*
        DEPART    SYSQ          Depart the system queue
*
 PDCT   TERMINATE 0             Terminate the XACT
*
*
*        Model Timing
*
*    Run model for 40 hour week
*
*
        GENERATE  60.*8.*5.
        TERMINATE 1
*
*
*********************************
*
*        GPSS/H Control Statements
*
*********************************
*
*        Process Time Variability (&DELT) initialized.
*        The values in successive models ranged between
*        0.0 and 0.40 in increments of 0.05.
*
*        Processing Time Variability = 0.00%
*
        LET       &DELT=0.0
*
*        Probability of Station Failure (&POF) initialiized.
*        The values in successive models ranged between
*        0.0 and 0.40 in increments of 0.05.
*
        LET       &POF=0.0
*
```

```
        LET         &BFSZ=1          Buffer size initialized
        DO          &J=1,10
BFFR    STORAGE     &BFSZ            Buffer storage set to &BFSZ
        DO          &I=1,100         100 iterations at each
*                                    set-point
        LET         &BUFF=0
        START       1,NP             40 hour warm-up period
        RESET
        START       4,NP             160 hour data gathering
*
*    Output of data to file
*
        PUTPIC      FILE=OUTA,(&DELT,&POF,N(PDCT),QM(SYSQ))
  ***.**  *.**  ****  ***
        CLEAR
        ENDDO
        LET         &BFSZ=&BFSZ+1
        ENDDO
        END
```

Appendix C: System 2 - JIT - GPSSH Program Listing

```
        REALLOCATE COM,150000
*
        SIMULATE
*
*       Variable Declaration Section
*
********************************************
*
*       Variable Definitions
*
********************************************
*
*       &I, &J  - Loop Counters
*       &BFSZ   - Defined Buffer Size
*       &BUFF1A - Buffer Counter for Buffer 1A
*       &BUFF1B - Buffer Counter for Buffer 1B
*       &BUFF2A - Buffer Counter for Buffer 2A
*       &BUFF2B - Buffer Counter for Buffer 2B
*       &BUFF3  - Buffer Counter for Buffer 3
*       &POF    - Defined Probability of Station Failure
*       &DELT   - Percent Variability of Processing Time
*       &MN1A   - Mean Processing Time for Station 1A
*       &MN1B   - Mean Processing Time for Station 1B
*       &MN2A   - Mean Processing Time for Station 2A
*       &MN2B   - Mean Processing Time for Station 2B
*       &MN3    - Mean Processing Time for Station 3
*       &MN4    - Mean Processing Time for Station 4
*       &MEAN1A - Calculated Actual Processing Time for Station 1A
*       &MEAN1B - Calculated Actual Processing Time for Station 1B
*       &MEAN2A - Calculated Actual Processing Time for Station 2A
*       &MEAN2B - Calculated Actual Processing Time for Station 2B
*       &MEAN3  - Calculated Actual Processing Time for Station 3
*       &MEAN4  - Calculated Actual Processing Time for Station 4
*       &MWIP   - Maximum Work-in-Process
*
********************************************
*
*
        INTEGER    &I
        INTEGER    &J
        INTEGER    &BFSZ,&BUFF1A,&BUFF1B,&BUFF2A,&BUFF2B,&BUFF3
        REAL       &POF,&MN1A,&MN1B,&MN2A,&MN2B,&MN3,&MN4
        REAL       &MEAN1A,&MEAN1B,&MEAN2A,&MEAN2B
        REAL       &MEAN3,&MEAN4,&DELT
        REAL       &MWIP
```

```
************************************

*

*       Declaration of Output File

*

************************************

*

*

 OUTA   FILEDEF     'THSJ2A1.OUT'

*

*       Variable Assignment Section

*

*

*************************************

*

*       Definition of Mean Processing Times

*

*************************************

*

*

        LET         &MN1A=7.5
        LET         &MN1B=5.0
        LET         &MN2A=12.5
        LET         &MN2B=7.5
        LET         &MN3=20.0
        LET         &MN4=5.0

*

*       GPSS/H Block Section

*

*************************************

*

*   Subcomponent A Assembly

*

*************************************

*

*

        GENERATE    0,,0            Generate a XACT whenever the
*                                   input buffer A is empty
        SEIZE       INBUFA          Seize input buffer A

*

*

*       Beginning of product flow

*

*

        TEST NE     &BUFF1A,&BFSZ    Test for room in Buffer 1A
        BLET        &BUFF1A=&BUFF1A+1 Increment Buffer counter
        TEST E      F(STA1A),0       Test status of Station 1A
        RELEASE     INBUFA           Release input buffer A

*

*

        QUEUE       SYSQ            Enter the system queue
*
        QUEUE       SUBAQ           Enter the subcomponent A queue
*
```

```
**********************************
*
*       Process 1A
*
**********************************
*
*
        SEIZE       STA1A               Seize Station 1A
        TEST LE     FRN(2),&POF,SRT1A   Test for failure of station
        FUNAVAIL    PRCS1A              Make process unavailable
        ADVANCE     2*&MN3              Repair delay time
        FAVAIL      PRCS1A              Make process available
*
 SRT1A  SEIZE       PRCS1A              Seize Process 1A
*
        BLET        &MEAN1A=&MN1A+(((FRN(3)-0.5)*2)*&MN1A*&DELT)
        ADVANCE     &MEAN1A
        RELEASE     PRCS1A
        RELEASE     STA1A               Release Station 1A
*
******************************
*
*       Buffer 1A
*
******************************
*
*
        ENTER       BFR1A               Enter the buffer
        TEST NE     &BUFF2A,&BFSZ       Test for room in next buffer
        BLET        &BUFF2A=&BUFF2A+1   Increment next buffer counter
        TEST E      F(STA2A),0          Test status of station
        BLET        &BUFF1A=&BUFF1A-1   Decrement present buffer counter
        LEAVE       FR1A                Leave buffer
*
************************************
*
*       Process 2A
*
************************************
*
        SEIZE       STA2A
        TEST LE     FRN(4),&POF,SRT2A
        FUNAVAIL    PRCS2A
        ADVANCE     2*&MN3
        FAVAIL      PRCS2A
*
 SRT2A  SEIZE       PRCS2A
*
        BLET        &MEAN2A=&MN2A+((FRN(5)-0.5)*2)*&MN2A*&DELT)
        ADVANCE     &MEAN2A
        RELEASE     PRCS2A
        RELEASE     STA2A
*
```

```
**************************
*
*      Buffer 2A
*
**************************
*
*
        ENTER       BFR2A                   Enter the buffer
        ADVANCE     0
        GATE LS     SUBA                    On "SET" go through gate
        LOGIC R     SUBA                    "RESET" gate
        BLET        &BUFF2A=&BUFF2A-1 Decrement Buffer counter
        LEAVE       BFR2A                   Leave buffer
*
        DEPART      SYSQ                    Leave System queue
*
        DEPART      SUBAQ                   Leave subcomponent A queue
*
        TERMINATE 0                         Terminate subcomponent A
*
*
************************************************
*
*   Subcomponent B Assembly
*
************************************************
*
*
        GENERATE    0,,0
        SEIZE       INBUFB
*
*
*
*   Beginning of product flow
*
*
        TEST NE     &BUFF1B,&BFSZ
        BLET        &BUFF1B=&BUFF1B+1
        TEST E      F(STA1B),0
        RELEASE     INBUFB
*
*
*
        QUEUE       SYSQ
*
        QUEUE       SUBBQ
*
*
*
*
```

```
********************************
*
*       Process 1B
*
********************************
*
*
        SEIZE       STA1B
        TEST LE     FRN(6),&POF,SRT1B
        FUNAVAIL    PRCS1B
        ADVANCE     2*&MN3
        FAVAIL      PRCS1B
*
 SRT1B  SEIZE       PRCS1B
*
        BLET        &MEAN1B=&MN1B+(((FRN(7)-0.5)*2)*&MN1B*&DELT)
        ADVANCE     &MEAN1B
        RELEASE     PRCS1B
        RELEASE     STA1B
*
****************************
*
*       Buffer 1B
*
****************************
*
*
        ENTER       BFR1B
        TEST NE     &BUFF2B,&BFSZ
        BLET        &BUFF2B=&BUFF2B+1
        TEST E      F(STA2B),0
        BLET        &BUFF1B=&BUFF1B-1
        LEAVE       BFR1B
********************************************
*
*       Process 2B
*
********************************************
*
*
        SEIZE       STA2B
        TEST LE     FRN(8),&POF,SRT2B
        FUNAVAIL    PRCS2B
        ADVANCE     2*&MN3
        FAVAIL      PRCS2B
*
 SRT2B  SEIZE       PRCS2B
*
        BLET        &MEAN2B=&MN2B+(((FRN(9)-0.5)*2)*&MN2B*&DELT)
        ADVANCE     &MEAN2B
        RELEASE     PRCS2B
        RELEASE     STA2B
*
```

```
**************************
*
*       Buffer 2B
*
**************************
*
*
*
        ENTER      BFR2B
        TEST NE    SE(BFR2A),1       Test for availability of
*                                    subcomponent A
*
        TEST NE    &BUFF3,&BFSZ      Test for room in next buffer
        BLET       &BUFF3=&BUFF3+1   Increment buffer 3 counter
        TEST E     F(STA3),0         Test status of Sta 3
        BLET       &BUFF2B=&BUFF2B-1 Decrement buffer counter
        LOGIC S    SUBA              "SET" subcomponent A gate
        LEAVE      BFR2B
*
        DEPART     SUBBQ
*
*********************************************************
*
*    Joining of Subcomponents A and B
*
*********************************************************
*
        QUEUE      ASSEMQ            Enter the assembly queue
*
*
*
************************************************
*
*       Process 3
*
************************************************
*
*
*
        SEIZE      STA3
*
        SEIZE      PRCS3
*
        BLET       &MEAN3=&MN3+(((FRN(10)-0.5)*2)*&MN3*&DELT)
        ADVANCE    &MEAN3
        RELEASE    PRCS3
        RELEASE    STA3
*
*
*
```

```
**************************
*
*       Buffer 3
*
**************************
*
*
        ENTER       BFFR3
        TEST E      F(STA4),0
        BLET        &BUFF3=&BUFF3-1
        LEAVE       BFFR3
*
*
*
*
************************************
*
*       Process 4
*
************************************
*
*
        SEIZE       STA4
        TEST LE     FRN(11),&POF,STRT4
        FUNAVAIL    PRCS4
        ADVANCE     2*&MN3
        FAVAIL      PRCS4
*
 STRT4  SEIZE       PRCS4
*
        BLET        &MEAN4=&MN4+(((FRN(12)-0.5)*2)*&MN4*&DELT)
        ADVANCE     &MEAN4
        RELEASE     PRCS4
        RELEASE     STA4
*
        DEPART      SYSQ
*
        DEPART      ASSEMQ
*
 PDCT   TERMINATE   0
*
*
*
*       Model Timing
*
*   Run model for 40 hour week
*
        GENERATE    60.*8.*5.
        TERMINATE   1
```

```
************************************

*
*       GPSS/H Control Statements
*
************************************

*
*
*       Process Time Variability (&DELT) initialized.
*       The values in successive models ranged between
*       0.0 and 0.40 in increments of 0.05.
*
*

        LET        &DELT=0.0
*
*       Probability of Station Failure (&POF) initialiized.
*       The values in successive models ranged between
*       0.0 and 0.40 in increments of 0.05.
*

        LET        &POF=0.0
*
        LET        &BFSZ=1        Buffer size initialized
*
        DO         &J=1,8
*
BFR1A  STORAGE     &BFSZ
BFR1B  STORAGE     &BFSZ
BFR2A  STORAGE     &BFSZ          Set all model buffers to
BFR2B  STORAGE     &BFSZ          same size
BFFR3  STORAGE     &BFSZ
*
        DO         &I=1,100       100 iterations at each
*                                 set-point
        LET        &BUFF1A=0
        LET        &BUFF1B=0
        LET        &BUFF2A=0      Initialize buffer counters
        LET        &BUFF2B=0
        LET        &BUFF3=0
        INITIAL    LR(SUBA)       Initialize Subcomponent A gate
*
        START      1,NP           40 hour warm-up
*
        RESET
*
        START      4,NP           160 hour data gathering
*
*       Calculate Maximum WIP in model
*
        LET        &MWIP=((0.5*QM(SUBAQ))+(0.5*QM(SUBBQ))+QM(ASSEMQ))
```

```
*
*       Output data to file
*
        PUTPIC      FILE=OUTA,(&DELT,&POF,N(PDCT),&MWIP)
 ***.**   *.**  ****  ****
        CLEAR
        ENDDO
        LET         &BFSZ=&BFSZ+1      Increment buffer size
        ENDDO
        END
```

```
        REALLOCATE COM,150000
*
        SIMULATE
*
*
*
*       Variable Declaration Section
*
*********************************************
*
*       Variable Definitions
*
*********************************************
*
*       &I, &J  - Loop Counters
*       &BFSZ   - Defined Buffer Size
*       &BUFF   - Buffer Counter for Buffer
*       &POF    - Defined Probability of  Station Failure
*       &DELT   - Percent Variability of  Processing Time
*       &MN1A   - Mean Processing Time for Station 1A
*       &MN1B   - Mean Processing Time for Station 1B
*       &MN2A   - Mean Processing Time for Station 2A
*       &MN2B   - Mean Processing Time for Station 2B
*       &MN3    - Mean Processing Time for Station 3
*       &MN4    - Mean Processing Time for Station 4
*       &MEAN1A - Calculated Actual Processing Time for Station 1A
*       &MEAN1B - Calculated Actual Processing Time for Station 1B
*       &MEAN2A - Calculated Actual Processing Time for Station 2A
*       &MEAN2B - Calculated Actual Processing Time for Station 2B
*       &MEAN3  - Calculated Actual Processing Time for Station 3
*       &MEAN4  - Calculated Actual Processing Time for Station 4
*       &MWIP   - Maximum Work-in-Process
*       &WIP    - Temporary inventory counter
*
*********************************************
*
*       Variable Declaration Section
*
        INTEGER    &I
        INTEGER    &J
        INTEGER    &BFSZ,&BUFF,&WIP,&MWIP
        REAL       &POF,&MN1A,&MN1B,&MN2A,&MN2B,&MN3,&MN4
        REAL       &MEAN1A,&MEAN1B,&MEAN2A,&MEAN2B
        REAL       &MEAN3,&MEAN4,&DELT
```

```
***********************..***************
*
*       Declaration of Output File
*
*******************************************
*
*
 OUTA   FILEDEF    'THSC2A1.OUT'
*
*
*       Variable Assignment Section
*
*******************************************
*
*       Definition of Mean Processing Times
*
*******************************************
*
        LET        &MN1A=7.5
        LET        &MN1B=5.0
        LET        &MN2A=12.5
        LET        &MN2B=7.5
        LET        &MN3=20.0
        LET        &MN4=5.0
        LET        &MWIP=0
*
*       GPSS/H Block Section
*
*******************************************
*
*    Subcomponent A Assembly
*
*******************************************
*
*
        GENERATE   0,,0            Generate a XACT whenever the
*                                  input buffer is empty
        SEIZE      INBUF           Seize the input buffer
*
*
        TEST NE    &BUFF,&BFSZ     Test for room in buffer
        BLET       &BUFF=&BUFF+1   Increment buffer counter
        RELEASE    INBUF           Leave the input buffer
*
*
        BLET       &WIP=&WIP+1     Increment the inventory
*                                  counter
*
        SPLIT      1,PRCB          Split the XACT in to 2
*                                  subcomponents
*
        TEST E     F(STA1A),0      Test status of Sta 1A
*
```

```
********************************
*
*       Process 1A
*
********************************
*
*
*
        SEIZE       STA1A               Enter the station
        TEST LE     FRN(2),&POF,SRT1A   Test for failure
        FUNAVAIL    PRCS1A              Make process unavailable
        ADVANCE     2*&MN3              Delay for repair
        FAVAIL      PRCS1A              Make process available
*
 SRT1A  SEIZE       PRCS1A              Enter processing
*
        BLET        &MEAN1A=&MN1A+(((FRN(3)-0.5)*2)*&MN1A*&DELT)
        ADVANCE     &MEAN1A
        RELEASE     PRCS1A
        RELEASE     STA1A               Leave the station and processing
*
        TEST E      F(STA2A),0          Test status of next station
*
*
*
********************************
*
*·      Process 2A
*
********************************
*
*
        SEIZE       STA2A
        TEST LE     FRN(4),&POF,SRT2A
        FUNAVAIL    PRCS2A
        ADVANCE     2*&MN3
        FAVAIL      PRCS2A
*
*
 SRT2A  SEIZE       PRCS2A
*
        BLET        &MEAN2A=&MN2A+(((FRN(5)-0.5)*2)*&MN2A*&DELT)
        ADVANCE     &MEAN2A
        RELEASE     PRCS2A
        RELEASE     STA2A
*
*
        TRANSFER    ,PDT
*
*
```

```
************************************

*

*    Subcomponent B Assembly

*

************************************

*

*

 PRCB  ADVANCE    0
        TEST E     F(STA1B),0     Test status of station

*

*

*

************************************

*

*       Process 1B

*

************************************

*

*

        SEIZE      STA1B
        TEST LE    FRN(6),&POF,SRT1B
        FUNAVAIL   PRCS1B
        ADVANCE    2*&MN3
        FAVAIL     PRCS1B

*

 SRT1B SEIZE      PRCS1B

*

        BLET       &MEAN1B=&MN1B+(((FRN(7)-0.5)*2)*&MN1B*&DELT)
        ADVANCE    &MEAN1B
        RELEASE    PRCS1B
        RELEASE    STA1B

*

        TEST E     F(STA2B),0
************************************

*

*       Process 2B

*

************************************

*

*

        SEIZE      STA2B
        TEST LE    FRN(8),&POF,SRT2B
        FUNAVAIL   PRCS2B
        ADVANCE    2*&MN3
        FAVAIL     PRCS2B

*

 SRT2B SEIZE      PRCS2B

*

        BLET       &MEAN2B=&MN2B+(((FRN(9)-0.5)*2)*&MN2B*&DELT)
        ADVANCE    &MEAN2B
        RELEASE    PRCS2B
        RELEASE    STA2B

*
```

```
********************************************
*
*     Joining of Subcomponents A and B
*
********************************************
*
*
*
 PDT    ADVANCE     0
*
        ASSEMBLE    2           Assemble the 2 subcomponents
*
*
*****************************
*
*     Buffer
*
*****************************
*
*
*
        ENTER       BFFR
        TEST E      F(STA3),0
        BLET        &BUFF=&BUFF-1
        LEAVE       BFFR
*
*
**********************************************
*
*     Process 3
*
**********************************************
*
*
*
        SEIZE       STA3
*
        SEIZE       PRCS3
*
        BLET        &MEAN3=&MN3+(((FRN(10)-0.5)*2)*&MN3*&DELT)
        ADVANCE     &MEAN3
        RELEASE     PRCS3
        RELEASE     STA3
*
        TEST E      F(STA4),0
*
*
```

```
*********************************

*

*       Process 4

*

*********************************

*

        SEIZE       STA4
        TEST LE     FRN(11),&POF,STRT4
        FUNAVAIL    PRCS4
        ADVANCE     2*&MN3
        FAVAIL      PRCS4

*

 STRT4  SEIZE       PRCS4

*

        BLET        &MEAN4=&MN4+(((FRN(12)-0.5)*2)*&MN4*&DELT)
        ADVANCE     &MEAN4
        RELEASE     PRCS4
        RELEASE     STA4

*

        TEST G      &WIP,&MWIP,BPSS     Test to see if current inventory
*                                       is greater than past
        BLET        &MWIP=&WIP
 BPSS   BLET        &WIP=&WIP-1         Decrement inventory counter

*

*

 PDCT   TERMINATE   0

*

*       Model Timing

*

*       Run model for 40 hour week

*

        GENERATE    60.*8.*5.
        TERMINATE   1

*

*********************************

*

*       GPSS/H Control Statements

*

*********************************

*

*       Process Time Variability (&DELT) initialized.
*       The values in successive models ranged between
*       0.0 and 0.40 in increments of 0.05.

*

        LET         &DELT=0.0

*

*       Probability of Station Failure (&POF) initialiized.
*       The values in successive models ranged between
*       0.0 and 0.40 in increments of 0.05.

*

        LET         &POF=0.0
```

```
*
        LET       &BFSZ=1          Initialize buffer size
        DO        &J=1,8
 BFFR   STORAGE   &BFSZ            Set buffer
*
        DO        &I=1,100         100 iterations at each
*                                  set-point
*
        LET       &BUFF=0          Initialize buffer counter
        LET       &WIP=0           Initialize inventory counter
        START     1,NP             40 hour warm-up
        RESET
        LET       &MWIP=0
        START     4,NP             160 hour data gathering
*
*       Output data to file
*
        PUTPIC    FILE=OUTA,(&DELT,&POF,N(PDCT),&MWIP)
 ***.***  *.***  *****  ****
        CLEAR
        ENDDO
        LET       &BFSZ=&BFSZ+1   Increment buffer size
        ENDDO
        END
```

Appendix E: System 3 - JIT - GPSSH Program Listing

```
        REALLOCATE COM,500000
*
        SIMULATE
*
*
*******************************************
*
*       Variable Definitions
*
*******************************************
*
*       &I, &J   - Loop Counters
*       &BFSZ    - Defined Buffer Size
*       &BUFF1A  - Buffer Counter for Buffer 1A
*       &BUFF1B  - Buffer Counter for Buffer 1B
*       &BUFF1C  - Buffer Counter for Buffer 1C
*       &BUFF2A  - Buffer Counter for Buffer 2A
*       &BUFF2B  - Buffer Counter for Buffer 2B
*       &BUFF2C  - Buffer Counter for Buffer 2C
*       &BF3X1   - Buffer Counter for Buffer 3-1
*       &BF3X2   - Buffer Counter for Buffer 3-2
*       &POF     - Defined Probability of Station Failure
*       &DELT    - Percent Variability of Processing Time
*       &MN1A    - Mean Processing Time for Station 1A
*       &MN1B    - Mean Processing Time for Station 1B
*       &MN1C    - Mean Processing Time for Station 1C
*       &MN2A    - Mean Processing Time for Station 2A
*       &MN2B    - Mean Processing Time for Station 2B
*       &MN2C    - Mean Processing Time for Station 2C
*       &MN3X1   - Mean Processing Time for Station 3-1
*       &MN3X2   - Mean Processing Time for Station 3-2
*       &MN4X1   - Mean Processing Time for Station 4-1
*       &MN4X2   - Mean Processing Time for Station 4-2
*       &MEAN1A  - Calculated Actual Processing Time for Station 1A
*       &MEAN1B  - Calculated Actual Processing Time for Station 1B
*       &MEAN1C  - Calculated Actual Processing Time for Station 1C
*       &MEAN2A  - Calculated Actual Processing Time for Station 2A
*       &MEAN2B  - Calculated Actual Processing Time for Station 2B
*       &MEAN2C  - Calculated Actual Processing Time for Station 2C
*       &MEAN3X1 - Calculated Actual Processing Time for Station 3-1
*       &MEAN3X2 - Calculated Actual Processing Time for Station 3-2
*       &MEAN4X1 - Calculated Actual Processing Time for Station 4-1
*       &MEAN4X2 - Calculated Actual Processing Time for Station 4-2
*       &MWIP    - Maximum Work-in-Process
*
*******************************************
*
```

```
*
*       Variable Declaration Section
*

        INTEGER     &I
        INTEGER     &J
        INTEGER     &BFSZ,&BUFF1A,&BUFF1B,&BUFF1C
        INTEGER     &BUFF2A,&BUFF2B,&BUFF2C
        INTEGER     &BF3X1,&BF3X2
        REAL        &POF,&MN1A,&MN1B,&MN1C,&MN2A,&MN2B,&MN2C
        REAL        &MN3X1,&MN3X2,&MN4X1,&MN4X2
        REAL        &MEAN1A,&MEAN1B,&MEAN1C,&MEAN2A,&MEAN2B,&MEAN2C
        REAL        &MEAN3X1,&MEAN3X2,&MEAN4X1,&MEAN4X2,&DELT
        REAL        &MWIP
*
*******************************************************
*
*       Declaration of Output File
*
*******************************************************
*
 OUTA   FILEDEF     'THSJ3A1.OUT'
*
*
*
*       Variable Assignment Section
*
*
*******************************************************
*
*       Definition of Mean Processing Times
*
*******************************************************
*
*
*
*

        LET         &MN1A=7.5
        LET         &MN1B=2.5
        LET         &MN1C=5.0
        LET         &MN2A=12.5
        LET         &MN2B=7.5
        LET         &MN2C=10.0
        LET         &MN3X1=20.0
        LET         &MN3X2=25.0
        LET         &MN4X1=5.0
        LET         &MN4X2=5.0
*
*       GPSS/H Block Section
*
*
*
```

```
************************************
*
*   Subcomponent A Assembly
*
************************************
*
*
        GENERATE   0,,0              Generate a XACT whenever there
*                                    is room in the input buffer
        SEIZE      INBUFA            Enter the input buffer
*
*
*
*   Beginning of product flow
*
*
        TEST NE    &BUFF1A,&BFSZ     Test for room in the first buffer
        BLET       &BUFF1A=&BUFF1A+1 Increment the buffer counter
        TEST E     F(STA1A),0        Test status of station
        RELEASE    INBUFA            Exit the input buffer
*
*
*
        QUEUE      SUBAQ             Enter the subassembly A queue
*
*
************************************
*
*   Process 1A
*
************************************
*
*
*
        SEIZE      STA1A             Enter Station 1A
        TEST LE    FRN(2),&POF,SRT1A Test for failure of process
        FUNAVAIL   PRCS1A            Make process unavailable
        ADVANCE    2*&MN1A           Repair delay
        FAVAIL     PRCS1A            Make process available
*
 SRT1A  SEIZE      PRCS1A            Begin processing
*
        BLET       &MEAN1A=&MN1A+(((FRN(3)-0.5)*2)*&MN1A*&DELT)
        ADVANCE    &MEAN1A
        RELEASE    PRCS1A
        RELEASE    STA1A             Release Station and process
```

E-3

```
***************************
*
*       Buffer 1A
*
***************************
*
        ENTER       BFR1A                   Enter Buffer 1A
        TEST NE     &BUFF2A,&BFSZ           Test for room in next buffer
        BLET        &BUFF2A=&BUFF2A+1       Increment buffer counter
        TEST E      F(STA2A),0              Test status of station
        BLET        &BUFF1A=&BUFF1A-1       Decrement the buffer counter
        LEAVE       BFR1A                   Leave the buffer
*
*
******************************************
*
*       Process 2A
*
******************************************
*
*
        SEIZE       STA2A
        TEST LE     FRN(4),&POF,SRT2A
        FUNAVAIL    PRCS2A
        ADVANCE     2*&MN2A
        FAVAIL      PRCS2A
*
 SRT2A  SEIZE       PRCS2A
*
        BLET        &MEAN2A=&MN2A+(((FRN(5)-0.5)*2)*&MN2A*&DELT)
        ADVANCE     &MEAN2A
        RELEASE     PRCS2A
        RELEASE     STA2A
*
***************************
*
*       Buffer 2A
*
***************************
*
        ENTER       BFR2A                   Enter the buffer
        TEST NE     SE(BFR2B),1             Check for Subcomponent B
*                                           availability
        TEST NE     &BF3X1,&BFSZ            Test for room in next buffer
        BLET        &BF3X1=&BF3X1+1         Increment counter
        TEST E      F(STA3X1),0             Test station status
        BLET        &BUFF2A=&BUFF2A-1       Decrement counter
        LOGIC S     SUBB                    "Set" Gate SUBB
        LEAVE       BFR2A                   Leave the buffer
*
        DEPART      SUBAQ                   Depart the queue
*
        TRANSFER    ,PDT1                   Transfer to Product 1 final assembly
```

```
********************************************
*
*    Subcomponent B Assembly
*
********************************************
*
*

        GENERATE    0,,0
        SEIZE       INBUFB
*
*
*
*    Beginning of product flow
*
*

        TEST NE     &BUFF1B,&BFSZ
        BLET        &BUFF1B=&BUFF1B+1
        TEST E      F(STA1B),0
        RELEASE     INBUFB
*
*
*

        QUEUE       SUBBQ
*
*
*
********************************************
*
*    Process 1B
*
********************************************
*
*

        SEIZE       STA1B
        TEST LE     FRN(6),&POF,SRT1B
        FUNAVAIL    PRCS1B
        ADVANCE     2*&MN1B
        FAVAIL      PRCS1B
*
 SRT1B  SEIZE       PRCS1B
*
        BLET        &MEAN1B=&MN1B+(((FRN(7)-0.5)*2)*&MN1B*&DELT)
        ADVANCE     &MEAN1B
        RELEASE     PRCS1B
        RELEASE     STA1B
```

```
****************************

*

*       Buffer 1B

*

****************************

*

        ENTER       BFR1B
        TEST NE      &BUFF2B,&BFSZ
        BLET         &BUFF2B=&BUFF2B+1
        TEST E       F(STA2B),0
        BLET         &BUFF1B=&BUFF1B-1
        LEAVE        BFR1B

*

***********************************

*

*       Process 2B

*

***********************************

*

*

        SEIZE       STA2B
        TEST LE     FRN(8),&POF,SRT2B
        FUNAVAIL    PRCS2B
        ADVANCE     2*&MN2B
        FAVAIL      PRCS2B

*

 SRT2B SEIZE        PRCS2B

*

        BLET        &MEAN2B=&MN2B+(((FRN(9)-0.5)*2)*&MN2B*&DELT)
        ADVANCE     &MEAN2B
        RELEASE     PRCS2B
        RELF^^?     STA2B

*

**********    ************

*

*       Buffer 2B

*

*****************************

*

*

        ENTER       BFR2B           Enter the buffer

*

        ADVANCE     0               Insure no blocking
        GATE LS     SUBB            Wait for "Set" of gate SUBB
        LOGIC R     SUBB            "Reset" the gate
        BLET        &BUFF2B=&BUFF2B-1 Decrement the buffer counter
        LEAVE       BFR2B           Leave the buffer

*

*

        DEPART      SUBBQ

*

        TERMINATE 0                 Terminate the transaction

*
```

```
************************************
*
*    Subcomponent C Assembly
*
************************************
*
       GENERATE    0,0,0
       SEIZE       INBUFC
*
*      Beginning of product flow
*
       TEST NE     &BUFF1C,&BFSZ
       BLET        &BUFF1C=&BUFF1C+1
       TEST E      F(STA1C),0
       RELEASE     INBUFC
*
*
       QUEUE       SUBCQ
*
*********************************
*
*      Process 1C
*
*********************************
*
*
       SEIZE       .STA1C
       TEST LE     FRN(10),&POF,SRT1C
       FUNAVAIL    PRCS1C
       ADVANCE     2*&MN1C
       FAVAIL      PRCS1C
*
 SRT1C SEIZE       PRCS1C
*
       BLET        &MEAN1C=&MN1C+(((FRN(11)~0.5)*2)*&MN1C*&DELT)
       ADVANCE     &MEAN1C
       RELEASE     PRCS1C
       RELEASE     STA1C
*
*****************************
*
*      Buffer 1C
*
*****************************
*
*
       ENTER       BFR1C
       TEST NE     &BUFF2C,&BFSZ
       BLET        &BUFF2C=&BUFF2C+1
       TEST E      F(STA2C),0
       BLET        &BUFF1C=&BUFF1C-1
       LEAVE       BFR1C
*
```

```
**********************************
*
*        Process 2C
*
**********************************
*
*
        SEIZE       STA2C
        TEST LE     FRN(12),&POf,SRT2C
        FUNAVAIL    PRCS2C
        ADVANCE     2*&MN2C
        FAVAIL      PRCS2C
*
 SRT2C  SEIZE       PRCS2C
*
        BLET        &MEAN2C=&MN2C+(((FRN(13)-0.5)*2)*&MN2C*&DELT)
        ADVANCE     &MEAN2C
        RELEASE     PRCS2C
        RELEASE     STA2C
*
***************************
*
*        Buffer 2C
*
***************************
*
*
        ENTER       BFR2C
        TEST NE     SE(BFR2B),1
        TEST NE     &BF3X2,&BFSZ
        BLET        &BF3X2=&BF3X2+1
        TEST E      F(STA3X2),0
        BLET        &BUFF2C=&BUFF2C-1
        LOGIC S     SUBB
        LEAVE       BFR2C
*
        DEPART      SUBCQ
*
        TRANSFER    ,PDT2
*
*********************************************
*
*     Joining of Subcomponents A and B
*
*********************************************
*
*
*
 PDT1   QUEUE       ASSEM1Q      Enter the assembly queue for Product 1
*
*
```

```
*********************************
*
*       Process 3 Product 1
*
*********************************
*
        SEIZE       STA3X1
*
        SEIZE       PRCS3X1
*
        BLET        &MEAN3X1=&MN3X1+(((FRN(14)-0.5)*2)*&MN3X1*&DELT)
        ADVANCE     &MEAN3X1
        RELEASE     PRCS3X1
        RELEASE     STA3X1
*
*****************************
*
*       Buffer 3-1
*
*****************************
*
        ENTER       BFR3X1
        TEST E      F(STA4X1),0
        BLET        &BF3X1=&BF3X1-1
        LEAVE       BFR3X1
*
*********************************
*
*       Process 4 Product 1
*
*********************************
*
        SEIZE       STA4X1
        TEST LE     FRN(15),&POF,SRT41
        FUNAVAIL    PRCS4X1
        ADVANCE     2*&MN4X1
        FAVAIL      PRCS4X1
*
 SRT4   SIZE        PRCS4X1
*
        BLET        &MEAN4X1=&MN4X1+(((FRN(16)-0.5)*2)*&MN4X1*&DELT)
        ADVANCE     &MEAN4X1
        RELEASE     PRCS4X1
        RELEASE     STA4X1
*
*
        DEPART      ASSEM1Q
*
 PDCT1  TERMINATE   0
```

```
**********************************
*
*    Joining of Subcomponents B and C
*
**********************************
*
PDT2   QUEUE      ASSEM2Q
*
**********************************
*
*      Process 3 Product 2
*
**********************************
*
       SEIZE      STA3X2
       SEIZE      PRCS3X2
*
       BLET       &MEAN3X2=&MN3X2+(((FRN(17)-0.5)*2)*&MN3X2*&DELT)
       ADVANCE    &MEAN3X2
       RELEASE    PRCS3X2
       RELEASE    STA3X2
*
**************************
*
*      Buffer 3-2
*
**************************
*
       ENTER      BFR3X2
       TEST E     F(STA4X2),0
       BLET       &BF3X2=&BF3X2-1
       LEAVE      BFR3X2
*
**********************************
*
*      Process 4 Product 2
*
**********************************
*
       SEIZE      STA4X2
       TEST LE    FRN(18),&POF,SRT42
       FUNAVAIL   PRCS4X2
       ADVANCE    2*&MN4X2
       FAVAIL     PRCS4X2
*
SRT42  SEIZE      PRCS4X2
       BLET       &MEAN4X2=&MN4X2+(((FRN(19)-0.5)*2)*&MN4X2*&DELT)
       ADVANCE    &MEAN4X2
       RELEASE    PRCS4X2
       RELEASE    STA4X2
       DEPART     ASSEM2Q
*
PDCT2  TERMINATE  0
```

```
*
*       Model Timing
*
*    Run model for 40 hour week
*
*
        GENERATE   60.*8.*5.
        TERMINATE  1
*
*
*
*********************************
*
*       GPSS/H Control Statements
*
*********************************
*
*
*       Process Time Variability (&DELT) initialized.
*       The values in successive models ranged between
*       0.0 and 0.40 in increments of 0.05.
*
        LET        &DELT=0.0
*
*       Probability of Station Failure (&POF, initialiized.
*       The values in successive models ranged between
*       0.0 and 0.40 in increments of 0.05.
*
        LET        &POF=0.0
*
        LET        &BFSZ=1         Buffer size initialized
*
        DO         &J=1,8
 BFR1A  STORAGE    &BFSZ
 BFR1B  STORAGE    &BFSZ
 BFR1C  STORAGE    &BFSZ           Set all model buffers
 BFR2A  STORAGE    &BFSZ           to the same size
 BFR2B  STORAGE    &BFSZ
 BFR2C  STORAGE    &BFSZ
 BFR3X1 STORAGE    &BFSZ
 BFR3X2 STORAGE    &BFSZ
        DO         &I=1,100        100 iterations at each
*                                  set-point
        LET        &BUFF1A=0
        LET        &BUFF1B=0
        LET        &BUFF1C=0
        LET        &BUFF2A=0       Initialize all buffer counters
        LET        &BUFF2B=0
        LET        &BUFF2C=0
        LET        &BF3X1=0
        LET        &BF3X2=0
        INITIAL    LR(SUBB)        Initialize the Gate SUBB
```

E-11

```
        START      1,NP          40 hour warm-up period
        RESET
        START      4,NP          160 hour data gathering
*
*       Calculation of Maximum Work-in-Process
*
        LET        &MWIP=((0.5*QM(SUBAQ))+(0.5*QM(SUBBQ))+_
                   (0.5*QM(SUBCQ))+QM(ASSEM1Q)+QM(ASSEM2Q))
*
*       Output data to file
*
        PUTPIC     FILE=OUTA,(&DELT,&POF,N(PDCT1),N(PDCT2),_
                   N(PDCT1)+N(PDCT2),&MWIP)
 ***.***   *.***  *****  *****  ******  *****
        CLEAR
        ENDDO
        LET        &BFSZ=&BFSZ+1   Increment buffer size
        ENDDO                      Repeat execution
        END
```

```
        REALLOCATE COM,500000
*
        SIMULATE
*
*
*
********************************************
*
*       Variable Definitions
*
********************************************
*
*       &I, &J  - Loop Counters
*       &BFSZ   - Defined Buffer Size
*       &BUFF1  - Buffer Counter for Buffer 1A
*       &BUFF2  - Buffer Counter for Buffer 2A
*       &POF    - Defined Probability of Station Failure
*       &DELT   - Percent Variability of Processing Time
*       &MN1A   - Mean Processing Time for Station 1A
*       &MN1B   - Mean Processing Time for Station 1B
*       &MN1C   - Mean Processing Time for Station 1C
*       &MN2A   - Mean Processing Time for Station 2A
*       &MN2B   - Mean Processing Time for Station 2B
*       &MN2C   - Mean Processing Time for Station 2C
*       &MN3X1  - Mean Processing Time for Station 3-1
*       &MN3X2  - Mean Processing Time for Station 3-2
*       &MN4X1  - Mean Processing Time for Station 4-1
*       &MN4X2  - Mean Processing Time for Station 4-2
*       &MEAN1A - Calculated Actual Processing Time for Station 1A
*       &MEAN1B - Calculated Actual Processing Time for Station 1B
*       &MEAN1C - Calculated Actual Processing Time for Station 1C
*       &MEAN2A - Calculated Actual Processing Time for Station 2A
*       &MEAN2B - Calculated Actual Processing Time for Station 2B
*       &MEAN2C - Calculated Actual Processing Time for Station 2C
*       &MEAN3X1- Calculated Actual Processing Time for Station 3-1
*       &MEAN3X2- Calculated Actual Processing Time for Station 3-2
*       &MEAN4X1- Calculated Actual Processing Time for Station 4-1
*       &MEAN4X2- Calculated Actual Processing Time for Station 4-2
*       &MWIP1  - Maximum Work-in-Process for Product 1
*       &WIP1   - Interim variable for calculating &MWIP1
*       &MWIP2  - Maximum Work-in-Process for Product 2
*       &WIP2   - Interim variable for calculating &MWIP2
*       &MWIP   - Total Work-in-Process for the model
*
********************************************
```

```
*
*
*       Variable Declaration Section
*
*

        INTEGER     &I
        INTEGER     &J
        INTEGER     &BFSZ,&BUFF1,&BUFF2
        INTEGER     &WIP1,&MWIP1,&WIP2,&MWIP2,&MWIP
        REAL        &POF,&MN1A,&MN1B,&MN1C,&MN2A,&MN2B,&MN2C
        REAL        &MN3X1,&MN3X2,&MN4X1,&MN4X2
        REAL        &MEAN1A,&MEAN1B,&MEAN1C,&MEAN2A,&MEAN2B,&MEAN2C
        REAL        &MEAN3X1,&MEAN3X2,&MEAN4X1,&MEAN4X2,&DELT
*
**********************************************
*
*       Declaration of Output File
*
**********************************************
*
*
 OUTA   FILEDEF     'THSC3A1.OUT'
*
*
*
*       Variable Assignment Section
*
*
*
**********************************************
*
*       Definition of Mean Processing Times
*
**********************************************
*
*
*
        LET         &MN1A=7.5
        LET         &MN1B=2.5
        LET         &MN1C=5.0
        LET         &MN2A=12.5
        LET         &MN2B=7.5
        LET         &MN2C=10.0
        LET         &MN3X1=20.0
        LET         &MN3X2=25.0
        LET         &MN4X1=5.0
        LET         &MN4X2=5.0
        LET         &MWIP1=0
        LET         &MWIP2=0
```

```
*
*       GPSS/H Block Section
*
*******************************************
*
*   Subcomponent A Assembly
*
*******************************************
*
*
        GENERATE   0,,0,,,1PH        Generate a transaction whenever
*                                    the input buffer has room
        SEIZE      INBUF1            Enter the input buffer 1
*
        ASSIGN     PROD,1,PH         Assign Product 1 code
*
*
*       Beginning of product flow
*
*
        TEST NE    &BUFF1,&BFSZ      Test for room in buffer
        BLET       &BUFF1=&BUFF1+1   Increment buffer counter
        TEST E     F(STA1A),0        Test status of station
        RELEASE    INBUF1            Exit the input buffer
*
        BLET       &WIP1=&WIP1+1     Increment the WIP counter
*
        SPLIT      1,PRCB            Split the transaction into
*                                   two subcomponents
*
*
*******************************************
*
*       Process 1A
*
*******************************************
*
*
        SEIZE      STA1A             Enter Station 1A
        TEST LE    FRN(2),&POF,SRT1A Test for failure
        FUNAVAIL   PRCS1A            Make process unavailable
        ADVANCE    2*&MN1A           Delay for repair of process
        FAVAIL     PRCS1A            Make process available
*
SRT1A   SEIZE      PRCS1A            Start processing
*
        BLET       &MEAN1A=&MN1A+(((FRN(3)-0.5)*2)*&MN1A*&DELT)
        ADVANCE    &MEAN1A
        RELEASE    PRCS1A
        RELEASE    STA1A             Leave station and process
*
        TEST E     F(STA2A),0        Test status of station
*
```

```
*********************************
*
*        Process 2A
*
*********************************
*
        SEIZE      STA2A
        TEST LE    FRN(4),&POF,SRT2A
        FUNAVAIL   PRCS2A
        ADVANCE    2*&MN2A
        FAVAIL     PRCS2A
*
 SRT2A  SEIZE      PRCS2A
*
        BLET       &MEAN2A=&MN2A+(((FRN(5)-0.5)*2)*&MN2A*&DELT)
        ADVANCE    &MEAN2A
        RELEASE    PRCS2A
        RELEASE    STA2A
*
        TRANSFER   ,PDT1                    Transfer to Product 1
*                                           assembly
*********************************************
*
*   Subcomponent B Assembly
*
*************************************************
*
 PRCB   ADVANCE    0       .
*
*        Beginning of product flow
*
        TEST E     F(STA1B),0
*
*********************************
*
*        Process 1B
*
*********************************
*
        SEIZE      STA1B
        TEST LE    FRN(6),&POF,SRT1B
        FUNAVAIL   PRCS1B
        ADVANCE    2*&MN1B
        FAVAIL     PRCS1B
*
 SRT1B  SEIZE      PRCS1B
*
        BLET       &MEAN1B=&MN1B+(((FRN(7)-0.5)*2)*&MN1B*&DELT)
        ADVANCE    &MEAN1B
        RELEASE    PRCS1B
        RELEASE    STA1B
*
        TEST E     F(STA2B),0
```

```
***********************************
*
*       Process 2B
*
***********************************
*
*
        SEIZE       STA2B
        TEST LE     FRN(8),&POF,SRT2B
        FUNAVAIL    PRCS2B
        ADVANCE     2*&MN2B
        FAVAIL      PRCS2B
*
 SRT2B  SEIZE       PRCS2B
*
        BLET        &MEAN2B=&MN2B+(((FRN(9)-0.5)*2)*&MN2B*&DELT)
        ADVANCE     &MEAN2B
        RELEASE     PRCS2B
        RELEASE     STA2B
*
*
*
        TEST E      PH(PROD),1,PDT2     Test if subcomponent is for
        TRANSFER    ,PDT1               Product 1 or 2
*
*
*************************************************
*
*   Subcomponent C Assembly
*
*************************************************
*
*
        GENERATE    0,,0,,,1PH
        SEIZE       INBUF2
*
*
        ASSIGN      PROD,2,PH
*
*   Beginning of product flow
*
*
        TEST NE     &BUFF2,&BFSZ
        BLET        &BUFF2=&BUFF2+1
        TEST E      F(STA1C),0
        RELEASE     INBUF2
*
        BLET        &WIP2=&WIP2+1
*
        SPLIT       1,PRCB
*
```

```
********************************
*
*       Prccess 1C
*
********************************
*
*
        SEIZE       STA1C
        TEST LE     FRN(10),&POF,SRT1C
        FUNAVAIL    PRCS1C
        ADVANCE     2*&MN1C
        FAVAIL      PRCS1C
*
 SRT1C SEIZE        PRCS1C
*
        BLET        &MEAN1C=&MN1C+(((FRN(11)-0.5)*2)*&MN1C*&DELT)
        ADVANCE     &MEAN1C
        RELEASE     PRCS1C
        RELEASE     STA1C
*
        TEST E      F(STA2C),0
*
*
********************************
*
*       Process 2C
*
********************************
*
*
        SEIZE       STA2C
        TEST LE     FRN(12),&POF,SRT2C
        FUNAVAIL    PRCS2C
        ADVANCE     2*&MN2C
        FAVAIL      PRCS2C
*
 SRT2C SEIZE        PRCS2C
*
        BLET        &MEAN2C=&MN2C+(((FRN(13)-0.5)*2)*&MN2C*&DELT)
        ADVANCE     &MEAN2C
        RELEASE     PRCS2C
        RELEASE     STA2C
*
*
        TRANSFER    ,PDT2
*
```

F-6

```
************************************************
*
*      Joining of Subcomponents A and B
*
************************************************
*
*
*
 PDT1  ADVANCE     0
       ASSEMBLE    2            Assemble the 2 subcomponents
*
*
*
******************************
*
*      Buffer 1
*
******************************
*
*
*
       ENTER       BFFR1        Enter the constraint buffer
       TEST E      F(STA3X1),0  Test station status
       BLET        &BUFF1=&BUFF1-1  Decrement buffer counter
       LEAVE       BFFR1        Leave the buffer
*
*
*
************************************************
*
*      Process 3 Product 1
*
************************************************
*
*
*
       SEIZE       STA3X1
*
       SEIZE       PRCS3X1
*
       BLET        &MEAN3X1=&MN3X1+(((FRN(14)-0.5)*2)*&MN3X1*&DELT)
       ADVANCE     &MEAN3X1
       RELEASE     PRCS3X1
       RELEASE     STA3X1
*
       TEST E      F(STA4X1),0
*
```

```
***********************************
*
*       Process 4 Product 1
*
***********************************
*
*
        SEIZE       STA4X1
        TEST LE     FRN(15),&POF,SRT41
        FUNAVAIL    PRCS4X1
        ADVANCE     2*&MN4X1
        FAVAIL      PRCS4X1
*
 SRT41 SEIZE        PRCS4X1
*
        BLET        &MEAN4X1=&MN4X1+(((FRN(16)-0.5)*2)*&MN4X1*&DELT)
        ADVANCE     &MEAN4X1
        RELEASE     PRCS4X1
        RELEASE     STA4X1
*
*
        TEST G      &WIP1,&MWIP1,BPSS1    Determine if current WIP
        BLET        &MWIP1=&WIP1          level is the maximum
 BPSS1 BLET         &WIP1=&WIP1-1         Decrement WIP counter
*
 PDCT1 TERMINATE    0
*
*
***********************************************
*
*    Joining of Subcomponents B and C
*
***************************************************
*
*
*
 PDT2  ADVANCE      0
        ASSEMBLE    2
*
*
***************************
*
*       Buffer 2
*
***************************
*
*
        ENTER       BFFR2
        TEST E      F(STA3X2),0
        BLET        &BUFF2=&BUFF2-1
        LEAVE       BFFR2
*
*
```

```
**********************************
*
*       Process 3 Product 2
*
**********************************
*
*
        SEIZE      STA3X2
*
        SEIZE      PRCS3X2
*
        BLET       &MEAN3X2=&MN3X2+(((FRN(17)-0.5)*2)*&MN3X2*&DELT)
        ADVANCE    &MEAN3X2
        RELEASE    PRCS3X2
        RELEASE    STA3X2
*
        TEST E     F(STA4X2),0
*
*
**********************************
*
*       Process 4 Product 2
*
**********************************
*
*
        SEIZE      STA4X2
        TEST LE    FRN(18),&POF,SRT42
        FUNAVAIL   PRCS4X2
        ADVANCE    2*&MN4X2
        FAVAIL     PRCS4X2
*
 SRT42  SEIZE      PRCS4X2
*
        BLET       &MEAN4X2=&MN4X2+(((FRN(19)-0.5)*2)*&MN4X2*&DELT)
        ADVANCE    &MEAN4X2
        RELEASE    PRCS4X2
        RELEASE    STA4X2
*
*
        TEST G     &WIP2,&MWIP2,BPSS2
        BLET       &MWIP2=&WIP2
 BPSS2  BLET       &WIP2=&WIP2-1
*
*
 PDCT2  TERMINATE  0
*
```

```
*
*       Model Timing
*
*    Run model for 40 hour week
*
         GENERATE   60.*8.*5.
         TERMINATE  1
*
*                                                                    •
*********************************
*
*       GPSS/H Control Statements
*
*********************************
*
*
*       Process Time Variability (&DELT) initialized.
*       The values in successive models ranged between
*       0.0 and 0.40 in increments of 0.05.
*
         LET          &DELT=0.0
*
*       Probability of Station Failure (&POF) initialiized.
*       The values in successive models ranged between
*       0.0 and 0.40 in increments of 0.05.    •
*
         LET          &POF=0.0
         LET          &BFSZ=1
         DO           &J=1,8
 BFFR1 STORAGE        &BFSZ          Initialize the buffers
 BFFR2 STORAGE        &BFSZ
         DO           &I=1,100       100 iterations at each set-point
         LET          &BUFF1=0       Initialize buffer counters
         LET          &BUFF2=0
         LET          &WIP1=0
         LET          &WIP2=0
         START        1,NP           40 hour warm-up
         RESET
         LET          &MWIP1=0
         LET          &MWIP2=0
         START        4.NP           160 hour data gathering
*
*       Calculation of Maximum Work-in-Process
*
         LET          &MWIP=&MWIP1+&MWIP2
```

```
*
*       Output data to file
*
        PUTPIC      FILE=OUTA,(&DELT,&POF,N(PDCT1),N(PDCT2),_
                    N(PDCT1)+N(PDCT2),&MWIP)
***.***  *.***  *****  *****  ******  *****
        CLEAR
        ENDDO
        LET         &BFSZ=&BFSZ+1
        ENDDO
        END
```

Bibliography

Austin, Wanda and Behrokh Khoshnevis. "A Model for Comparison of the Traditional Push Type and the Just in Time Production Configurations," Proceedings of the Conference on Modeling and Simulation on Microcomputers. 191-196 (1986).

Banks, Jerry and others. Getting Started with GPSS/H. Annadale VA: Wolverine Software Corporation, 1989.

Berry, Bryan H. "Production's New Outer Limit," Iron Age, 228: 81-82 (August 2, 1985).

Bowen, H. L. "GPSS Study of Work-in-Process Inventory" Proceedings of the 4th Conference on Applications of Simulations. 162-165 (1970).

Deming, W. Edwards. Out of the Crisis. Cambridge MA: Massachusetts Institute of Technology Center for Advanced Engineering Study, 1982.

Devore, Jay L. Probability and Statistics for Engineering and the Sciences (Second Edition). Monterey CA: Brooks/Cole Publishing Co., 1987.

Dunham, N. R. and A. K. Kochhar. "Approaches to the Computer Simulation of Production Systems," Proceedings of the Conference on Simulation in Inventory and Production Control. San Diego CA: Society of Computer Simulation, 1983.

Goldratt, Eliyahu M. The Haystack Syndrome. Croton-on-Hudson NY: North River Press, 1990a.

------, Theory of Constraints. Croton-on-Hudson NY: North River Press, 1990b.

------ and Robert E. Fox. "How Complex are Our Systems," The Theory of Constraints Journal, 1:5 1-14 (July/August 1989a).

------ and Robert E. Fox. "Looking Beyond the First Stage; Just In Time: Part 2," The Theory of Constraints Journal, 1:5 15-48 (July/August 1989b).

------ and Robert E. Fox. The Race. Croton-on-Hudson NY: North River Press, 1986.

------ and Jeff Cox. The Goal: A Process of Ongoing
        Improvement (Revised Edition). Croton-on-Hudson NY:
        North River Press, 1986.

Heiko, Lance. "A Simple Framework for Understanding JIT,"
        Production and Inventory Management Journal, 22: 61-63
        (Fourth Quarter 1989).

Huang, Philip Y. and others. "A Simulation Analysis of the
        Japanese Just-in-Time Technique (With Kanbans) for a
        Multiline, Multistage Production System," Decision
        Sciences, 14: 326-344 (July 1983).

Jayson, Susan. "Goldratt & Fox: Revolutionizing the
        Factory Floor," Management Accounting, 68: 18-22 (May
        1987).

Karmarkar, Uday S. Push, Pull, and Hybrid Control Schemes.
        William E. Simon Graduate School of Business
        Administration, Working Paper Series No. QM 86-14,
        University of Rochester, Rochester NY (1986).

Ketcham, Michael G. "Simulation Modeling of Just-in-Time
        Assembly Systems Using an Information-based
        Methodology," Proceedings of the 1988 Winter Simulation
        Conference. 624-628. New York: Association for
        Computing Machinery, 1988.

Lansburgh, Richard H. "Control in Continuous, Intermittent
        and Variable Processing," Scientific Management in
        American Industry, The Taylor Society, edited by Harlow
        S. Person. New York: Harper & Brothers, 1929.

Law, Averill M. "Simulation of Manufacturing Systems,"
        Proceedings of the 1988 Winter Simulation Conference.
        New York: Association of Computing Machinery, 1988.

Lulu, Menberu. "Just-in-time (JIT) Production and Process
        Unreliability," Proceedings of the 19th Annual IEEE
        Simulation Symposium. 237-249. Silver Springs MD: IEEE
        Computer Society Press, 1986.

Lulu, Menberu and J. T. Black. "Just-in-Time (JIT)
        Production and Process Cycle Time Variability,"
        Proceedings of the 20th Annual IEEE Simulation
        Symposium. 215-229. Silver Springs MD: IEEE Computer
        Society Press, 1987.

Malcolm, Donald G. "System Simulation - A Fundamental Tool
        for Industrial Engineering," The Journal of Industrial
        Engineering, 9:3 177-187 (May-June 1958).

Manivannan, S. and C. Dennis Pegden. "A Rule-Based Simulator for Modeling Just-in-Time Manufacturing Systems (JITSAI)," _Simulation, 55:_ 109-117 (August 1990).

McGurrin, Lisa. "Consultant's Credo: Success May Require Slaughtering the Sacred Cows of Business," _New England Business, 9:_ 37-38 (March 16, 1987).

Panwalker, S. S. and Wafik Iskander. "A Survey of Scheduling Rules," _Operations Research, 25:1_ 45-61 (January-February 1977).

Pyke, David F. and Morris A. Cohen. "Push and Pull in Manufacturing and Distribution Systems," _Journal of Operations Management, 9:1_ 24-43 (January 1990).

Ricard, Leonard J. "Fundamental Concepts of Just In Time," _APICS Seminar Proceedings - Zero Inventory Philosophy and Practices._ 1-9 (October 1984).

Rowe, Alan J. "Toward a Theory of Scheduling," _The Journal of Industrial Engineering, 11:2_ 125-136 (March-April 1960).

Russell, John R. "Simulation for Production," _Harvard Business Review, 45:_ 162-170 (September-October 1967).

Schonberger, Richard J. _Japanese Manufacturing Techniques._ New York: The Free Press, 1982.

Schonberger, Richard J. and Edward M. Knod, Jr. _Operations Management: Improving Customer Service_ (Fourth Edition). Boston MA: Richard D. Irwin, Inc., 1991.

Schroer, Bernard J. and others. "Just-in-Time, with Kanban, Manufacturing System Simulation on a Microcomputer," _Simulation, 45:2_ 62-70 (August 1985).

Severs, Capt Jefferson L. _The DISASTER Scheduling System: A Review and Case Analysis._ MS Thesis, AFIT/GLM/LSM/91S-56. School of Systems and Logistics, Air Force Institute of Technology (AU), Wright Patterson AFB OH, September 1991.

Skousen and others. _Accounting: Principles and Applications_ (Third Edition). New York: Worth Publishers, 1987.

Spearman, Mark L. and Michael A. Zazanis. _Push and Pull Production Systems: Issues and Comparisons._ Department of Industrial Engineering and Management Sciences, Technical Report 88-24, Northwestern University, Evanston IL (1988).

Suzaki, Kiyoshi. The New Manufacturing Challenge. New York: The Free Press, 1987.

Wallace, Thomas F. and John R. Dougherty. APICS Dictionary (Sixth Edition). Falls Church VA: American Production and Inventory Control Society, 1987.

Walton, Mary. The Deming Management Method. New York: The Putnam Publishing Group, 1986.

Waters, Craig R. "One of a Kind," Inc., 8: 107-108 (July 1986).

Wolverine Software Corp. GPSS/H VAX/VMS Release 2.0 (OV069). Annandale VA, 1990.

Vita

Captain Lynn A. Sines was born on 29 August 1961 in Indianapolis, Indiana. He graduated from Lewis Cass Jr-Sr High School in Walton, Indiana in 1979 and attended Purdue University, graduating with a Bachelor of Science in Electrical Engineering (specialty: Control Systems) in December 1983. Upon graduation, he attended Officer Training School and was commissioned in the USAF in March 1984. He served his first tour of duty at Tinker AFB, Oklahoma as a Technical Activities Management Engineer in the Directorate of Maintenance, Oklahoma City Air Logistics Center where he was responsible for the Electrical Engineering support of accessory items for all USAF aircraft. In February 1987 he was transferred to Wright Patterson AFB, Ohio where he was assigned as a Logistics Plans Officer in the Air Force Contract Maintenance Center (AFCMC) where he was responsible for the initial planning of world-wide contract administration activities. In June 1989 he was reassigned as Chief, Plans and Programs Division for AFCMC. There he was responsible for the planning of support for overseas coproduction and offset programs. He entered the School of Systems and Logistics, Air Force Institute of Technology, in May 1990.

Permanent Address: 2500 Sherman Dr.
Kokomo, IN 46902

REPORT DOCUMENTATION PAGE

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE September 1991	3. REPORT TYPE AND DATES COVERED Master's Thesis

4. TITLE AND SUBTITLE
COMPARISON OF THE OPERATIONAL CHARACTERISTICS OF THE THEORY OF CONSTRAINTS AND JUST-IN-TIME SCHEDULING METHODOLOGIES

5. FUNDING NUMBERS

6. AUTHOR(S)

Lynn A. Sines, Captain, USAF

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology, WPAFB OH 45433-6583

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/GSM/LSM/91S-24

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

This study compared the characteristics of scheduling using two approaches: Just-in-Time (JIT) and the Theory of Constraints (TOC). Computer simulation was used to evaluate changes in the throughput of the systems due to system variability and varying work-in-process (WIP) levels. The independent variables were processing time, probability of failure, and WIP level. The literature search revealed that these variables impacted performance of JIT systems. Simulation models of three different production systems were developed to determine if the TOC scheduled system would provide greater or equivalent throughput for equivalent levels of WIP, and to determine if the system production flow path had an impact on the performance of the TOC system. The models were scheduled using both JIT and TOC methodologies. The mean throughput for each combination of independent variables was calculated along with the 95 percent confidence interval. The mean and confidence intervals were then graphed for analysis. Simulation results indicated that the TOC system outperforms the JIT system for single product systems while the results of the multiple product system were inconclusive. There was no difference between the TOC systems for process time variability, but there was a difference for process station failure.

14. SUBJECT TERMS Production, Simulation, Scheduling, Just-in-Time, Theory of Constraints	15. NUMBER OF PAGES 160
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL

NSN 7540-01-280-5500

AFIT RESEARCH ASSESSMENT

The purpose of this questionnaire is to determine the potential for current and future applications of AFIT thesis research. Please return completed questionnaires to: AFIT/LSC, Wright-Patterson AFB OH 45433-6583.

1. Did this research contribute to a current research project?

    a. Yes   b. No

2. Do you believe this research topic is significant enough that it would have been researched (or contracted) by your organization or another agency if AFIT had not researched it?

    a. Yes   b. No

3. The benefits of AFIT research can often be expressed by the equivalent value that your agency received by virtue of AFIT performing the research. Please estimate what this research would have cost in terms of manpower and/or dollars if it had been accomplished under contract or if it had been done in-house.

    Man Years _____ $_____

4. Often it is not possible to attach equivalent dollar values to research, although the results of the research may, in fact, be important. Whether or not you were able to establish an equivalent value for this research (3 above), what is your estimate of its significance?

    a. Highly   b. Significant   c. Slightly   d. Of No
        Significant   Significant   Significance

5. Comments

_____   _____
Name and Grade   Organization

_____   _____
Position or Title   Address